Compendio de Silabos

Escuela Profesional de Ciencia de la Computación

– 2018-II–

Lima: 4 de enero de 2019

Equipo de trabajo

Comisión de Evaluación Curricular

Oscar Benito Pacheco
Docente de la E.A.P. de Computación Científica, UNMSM, Lima
email: *obenitop@unmsm.edu.pe*

Hugo Vega Huerta
Docente de la E.A.P. de Ingeniería de Sistemas, UNMSM, Lima
email: *hugovegahuerta@hotmail.com*

Silverio Bustos
Docente de la E.A.P. de Ingeniería de Sistemas, UNMSM, Lima
email: *silverio.bustos@gmail.com*

Luis Rivera Escriba
Docente de la Universidade Estadual do Norte Fluminense, Brasil
email: *rivera@uenf.br*

Ulises Román Concha
Docente de la E.A.P. de Ingeniería de Sistemas, UNMSM, Lima
email: *nromanc@yahoo.es*

Cristian Loli Prudencio
Docente de la E.A.P. de Ingeniería de Sistemas, UNMSM, Lima
email: *cristianhz@hotmail.com*

Además, han colaborado con este esfuerzo la Sociedad Peruana de Computación a quienes dejamos público nuestro agradecimiento.

Índice general

First Semester **5**

1.1. CS1100. Introduction to Computer Science 5
1.2. CS1D01. Discrete Structures I 11
1.3. GH0005. Communication Laboratory I 15
1.4. EG0003. Mathematics I 17
1.5. EG0004. Global Challenges 22

Second Semester **24**

2.1. CS1102. Objects oriented programming I 24
2.2. CS1D02. Discrete Structures II 33
2.3. ME0019. Physics I 37
2.4. GH0006. Communication Laboratory II 42
2.5. GH0007. Introduction to Business Development 44
2.6. GH1002. Art and Technology 47
2.7. GH1101. English I 50
2.8. EG0005. Math II 54

Third Semester **58**

3.1. CS1103. Objects oriented programming II 58
3.2. CS2201. Computer Architecture 69
3.3. CS2B01. Platform Based Development 77
3.4. EN0021. Physics II 82
3.5. GH0008. Business Management 87
3.6. GH1102. English II 89
3.7. EG0006. Math III 93

Fourth Semester **96**

4.1. CS2100. Algorithms and Data Structures 96
4.2. CS2101. Theory of Computation 99
4.3. CS2701. Databases I 103
4.4. IN0054. Statistics and Probabilities 109
4.5. GH2101. English II 112

Fifth Semester **116**

5.1. CS2102. Analysis and Design of Algorithms 116
5.2. CS2702. Databases II 123
5.3. CS2901. Software Engineering I 128
5.4. CS2S01. Operating systems 133

5.5. CS3402. Compilers 141
5.6. GH0010. Ethics and Technology 146
5.7. ID104. English IV 149

Sixth Semester 153
6.1. CS2301. Networking and Communication 153
6.2. CS3101. Competitive Programming 158
6.3. CS3102. Advanced Data Structures 160
6.4. CS3903. Information systems 164

Seventh Semester 167
7.1. CS2H01. Computer Human Interaction 167
7.2. CS341. Programming languages 175
7.3. CS3P01. Parallel and Distributed Computing 182
7.4. CS2501. Computer graphics 189
7.5. CS2601. Artificial intelligence 196
7.6. CS2902. Software Engineering II 205

Eighth Semester 212
8.1. CS4002. Capstone Project I 212
8.2. ET201. Entrerpreneurship I 215

Ninth Semester 220
9.1. CS3700. Big Data 220
9.2. CS3I01. Computer Security 224
9.3. CS4003. Final Project II 234
9.4. CS3501. Topics in Computer Graphics 237
9.5. CS3602. Robotics 239
9.6. CS3901. Software Engineering III 243
9.7. BI0021. Bioinformatics and Biostatistics 250
9.8. ET301. Entrerpreneurship II 254

Tenth Semester 258
10.1. CS3P02. Cloud Computing 258
10.2. CS3P03. Internet of Things 265
10.3. CS4004. Final Project III 272

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS1100. Introduction to Computer Science (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. Bibliography

[Bro11] J. Glenn Brookshear. *Computer Science: An Overview.* Addison-Wesley, 2011.

[Gut13] John V Guttag. . *Introduction To Computation And Programming Using Python.* MIT Press, 2013.

[Zel10] John Zelle. *Python Programming: An Introduction to Computer Science.* Franklin, Beedle & Associates Inc, 2010.

6. Information about the course

(a) **Brief description about the course** This is the first course in the sequence of introductory courses to Computer Science.This course is intended to cover the concepts outlined by the Computing Curricula IEEE-CS/ACM 2013. Programming is one of the pillars of Computer Science; any professional of the area, will need to program to materialize their models and proposals. This course introduces participants to the fundamental concepts of this art. Topics include data types, control structures, functions, lists, recursion, and the mechanics of execution, testing, and debugging.

(b) **Prerrequisites:** None

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Introduce the fundamental concepts of programming.
- Develop the ability of abstraction using programming language

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome b**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome b**

10. List of topics

1. History
2. Basic Type Systems
3. Fundamental Programming Concepts
4. Basic Analysis
5. Fundamental Data Structures and Algorithms
6. Algorithms and Design
7. Development Methods

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: History (5)	
Competences Expected: C4	
Learning Outcomes	**Topics**
• Identify significant continuing trends in the history of the computing field [Familiarity] • Identify the contributions of several pioneers in the computing field [Familiarity] • Discuss the historical context for several programming language paradigms [Familiarity] • Compare daily life before and after the advent of personal computers and the Internet [Assessment]	• Prehistory, the world before 1946 • History of computer hardware, software, networking • Pioneers of computing • History of the Internet
Readings : [Bro11], [Gut13], [Zel10]	

Unit 2: Basic Type Systems (2)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• For both a primitive and a compound type, informally describe the values that have that type [Familiarity] • For a language with a static type system, describe the operations that are forbidden statically, such as passing the wrong type of value to a function or method [Familiarity] • Describe examples of program errors detected by a type system [Familiarity] • For multiple programming languages, identify program properties checked statically and program properties checked dynamically [Usage] • Use types and type-error messages to write and debug programs [Usage] • Define and use program pieces (such as functions, classes, methods) that use generic types, including for collections [Usage]	• A type as a set of values together with a set of operations – Primitive types (e.g., numbers, Booleans) – Compound types built from other types (e.g., records, unions, arrays, lists, functions, references) • Association of types to variables, arguments, results, and fields • Type safety and errors caused by using values inconsistently given their intended types
Readings : [Gut13], [Zel10]	

Unit 3: Fundamental Programming Concepts (9)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Analyze and explain the behavior of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion. [Assessment] • Identify and describe uses of primitive data types [Familiarity] • Write programs that use primitive data types [Usage] • Modify and expand short programs that use standard conditional and iterative control structures and functions [Usage] • Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing [Usage] • Write a program that uses file I/O to provide persistence across multiple executions [Usage] • Choose appropriate conditional and iteration constructs for a given programming task [Familiarity] • Describe the concept of recursion and give examples of its use [Assessment] • Identify the base case and the general case of a recursively-defined problem [Familiarity]	• Basic syntax and semantics of a higher-level language • Variables and primitive data types (e.g., numbers, characters, Booleans) • Expressions and assingments • Simple I/O including file I/O • Conditional and iterative control structures • Functions and parameter passing • The concept of recursion
Readings : [Gut13], [Zel10]	

Unit 4: Basic Analysis (2)	
Competences Expected: C1,C5	
Learning Outcomes	**Topics**
• Explain what is meant by "best", "expected", and "worst" case behavior of an algorithm [Familiarity] • In the context of specific algorithms, identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors [Familiarity] • State the formal definition of big O [Familiarity] • Use big O notation formally to give asymptotic upper bounds on time and space complexity of algorithms [Usage] • Use big O notation formally to give expected case bounds on time complexity of algorithms [Usage]	• Differences among best, expected, and worst case behaviors of an algorithm • Big O notation: formal definition • Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential • Big O notation: use • Analysis of iterative and recursive algorithms
Readings : [Gut13], [Zel10]	

Unit 5: Fundamental Data Structures and Algorithms (8)	
Competences Expected: C1,C2,C5	
Learning Outcomes	**Topics**
• Implement basic numerical algorithms [Usage] • Implement simple search algorithms and explain the differences in their time complexities [Assessment] • Be able to implement common quadratic and O(N log N) sorting algorithms [Usage] • Describe the implementation of hash tables, including collision avoidance and resolution [Familiarity] • Discuss the runtime and memory efficiency of principal algorithms for sorting, searching, and hashing [Familiarity] • Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Familiarity] • Explain how tree balance affects the efficiency of various binary search tree operations [Familiarity] • Solve problems using fundamental graph algorithms, including depth-first and breadth-first search [Usage] • Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context [Assessment] • Describe the heap property and the use of heaps as an implementation of priority queues [Familiarity] • Solve problems using graph algorithms, including single-source and all-pairs shortest paths, and at least one minimum spanning tree algorithm [Usage] • Trace and/or implement a string-matching algorithm [Usage]	• Simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max, • Sequential and binary search algorithms • Worst case quadratic sorting algorithms (selection, insertion) • Worst or average case O(N log N) sorting algorithms (quicksort, heapsort, mergesort) • Hash tables, including strategies for avoiding and resolving collisions • Binary search trees – Common operations on binary search trees such as select min, max, insert, delete, iterate over tree • Graphs and graph algorithms – Representations of graphs (e.g., adjacency list, adjacency matrix) – Depth- and breadth-first traversals • Heaps • Graphs and graph algorithms – Shortest-path algorithms (Dijkstra's and Floyd's algorithms) – Minimum spanning tree (Prim's and Kruskal's algorithms) • Pattern matching and string/text algorithms (e.g., substring matching, regular expression matching, longest common subsequence algorithms)
Readings : [Gut13], [Zel10]	

Unit 6: Algorithms and Design (9)	
Competences Expected: C1,C2,C5	
Learning Outcomes	**Topics**
• Discuss the importance of algorithms in the problem-solving process [Familiarity] • Discuss how a problem may be solved by multiple algorithms, each with different properties [Familiarity] • Create algorithms for solving simple problems [Usage] • Use a programming language to implement, test, and debug algorithms for solving simple problems [Usage] • Implement, test, and debug simple recursive functions and procedures [Usage] • Determine whether a recursive or iterative solution is most appropriate for a problem [Assessment] • Implement a divide-and-conquer algorithm for solving a problem [Usage] • Apply the techniques of decomposition to break a program into smaller pieces [Usage] • Identify the data components and behaviors of multiple abstract data types [Usage] • Implement a coherent abstract data type, with loose coupling between components and behaviors [Usage] • Identify the relative strengths and weaknesses among multiple designs or implementations for a problem [Assessment]	• The concept and properties of algorithms – Informal comparison of algorithm efficiency (e.g., operation counts) • The role of algorithms in the problem-solving process • Problem-solving strategies – Iterative and recursive mathematical functions – Iterative and recursive traversal of data structures – Divide-and-conquer strategies • Fundamental design concepts and principles – Abstraction – Program decomposition – Encapsulation and information hiding – Separation of behaivor and implementation
Readings : [Gut13], [Zel10]	

Unit 7: Development Methods (1)	
Competences Expected: C2	
Learning Outcomes	**Topics**
• Construct and debug programs using the standard libraries available with a chosen programming language [Familiarity]	• Modern programming enviroments – Code search – Programming using library components and their APIs
Readings : [Gut13], [Zel10]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS1D01. Discrete Structures I (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HP; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Epp10] Susanna S. Epp. *Discrete Mathematics with Applications.* 4 ed. Brooks Cole, 2010.

[Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction.* 5 ed. Pearson, 2003.

[Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications.* 7 ed. Mc Graw Hill, 2007.

[Sch12] Edward R. Scheinerman. *Mathematics: A Discrete Introduction.* 3 ed. Brooks Cole, 2012.

6. **Information about the course**

(a) **Brief description about the course** Discrete structures provide the theoretical foundations necessary for computation. These fundamentals are not only useful to develop computation from a theoretical point of view as it happens in the course of computational theory, but also is useful for the practice of computing; In particular in applications such as verification, cryptography, formal methods, etc.

(b) **Prerrequisites:** None

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Apply Properly concepts of finite mathematics (sets, relations, functions) to represent data of real problems.
- Model real situations described in natural language, using propositional logic and predicate logic.
- Determine the abstract properties of binary relations.
- Choose the most appropriate demonstration method to determine the veracity of a proposal and construct correct mathematical arguments.
- Interpret mathematical solutions to a problem and determine their reliability, advantages and disadvantages.
- Express the operation of a simple electronic circuit using Boolean algebra.

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome j**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome j**

10. List of topics

1. Sets, Relations, and Functions
2. Basic Logic
3. Proof Techniques
4. Data Representation

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Sets, Relations, and Functions (22)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Explain with examples the basic terminology of functions, relations, and sets [Assessment] • Perform the operations associated with sets, functions, and relations [Assessment] • Relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context [Assessment]	• Sets – Venn diagrams – Union, intersection, complement – Cartesian product – Power sets – Cardinality of finite sets • Relations: – Reflexivity, simmetry, transitivity – Equivalence relations – Partial order relations and sets – Extremal elements of a partially ordered sets • Functions – Surjections, injections, bijections – Inverses – Composition
Readings : [Gri03], [Ros07]	

Unit 2: Basic Logic (14)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Convert logical statements from informal language to propositional and predicate logic expressions [Usage] • Apply formal methods of symbolic propositional and predicate logic, such as calculating validity of formulae and computing normal forms [Usage] • Use the rules of inference to construct proofs in propositional and predicate logic [Usage] • Describe how symbolic logic can be used to model real-life situations or applications, including those arising in computing contexts such as software analysis (eg, program correctness), database queries, and algorithms [Familiarity] • Apply formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles [Usage] • Describe the strengths and limitations of propositional and predicate logic [Usage]	• Propositional logic • Logical connectives • Truth tables • Normal forms (conjunctive and disjunctive) • Validity of well-formed formula • Propositional inference rules (concepts of modus ponens and modus tollens) • Predicate logic – Universal and existential quantification • Limitations of propositional and predicate logic (e.g., expressiveness issues)
Readings : [Ros07], [Gri03]	

Unit 3: Proof Techniques (14)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Identify the proof technique used in a given proof [Assessment] • Outline the basic structure of each proof technique (direct proof, proof by contradiction, and induction) described in this unit [Usage] • Apply each of the proof techniques (direct proof, proof by contradiction, and induction) correctly in the construction of a sound argument [Usage] • Determine which type of proof is best for a given problem [Assessment] • Explain the parallels between ideas of mathematical and/or structural induction to recursion and recursively defined structures [Familiarity] • Explain the relationship between weak and strong induction and give examples of the appropriate use of each [Assessment] • State the well-ordering principle and its relationship to mathematical induction [Familiarity]	• Notions of implication, equivalence, converse, inverse, contrapositive, negation, and contradiction • The structure of mathematical proofs • Direct proofs • Disproving by counterexample • Proof by contradiction • Induction over natural numbers • Structural induction • Weak and strong induction (i.e., First and Second Principle of Induction) • Recursive mathematical definitions • Well orderings
Readings : [Ros07], [Epp10], [Sch12]	

Unit 4: Data Representation (10)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Explain numerical representations such as sign-magnitude and floating point. [Assessment]. • Carry out arithmetic operations using different kinds of representations. [Assessment]. • Explain the floating point standard IEEE-754 [Familiarity].	• Numerical representation: sign-magnitude, floating point. • Representation of other objects: sets, relations, functions.
Readings : [Ros07], [Gri03]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: GH0005. Communication Laboratory I (Mandatory)
2. Credits: 3
3. Hours of theory and Lab: 2 HT; 2 HP; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[D93] Cassany. D. *La cocina de la Escritura*. Barcelona,España,Anagrama, 1993.

6. Information about the course

(a) **Brief description about the course** Through this course, the student will improve and strengthen his abilities to communicate both oral and written in an academic context. To do this, the student will exercise in the composition of texts, taking into account the requirements of a formal academic language: characteristics of academic writing (rules of punctuation, spelling, grammatical lexical competence, normative) and correct use of information. In turn, the course promotes a comprehensive reading that is not limited to the descriptive level, but also encompasses the conceptual and metaphorical, because only in this way will the student develop his critical and analytical capacity. The student will take on academic readings and scientific outreach that will allow him to distinguish between the objectives set out in the different types of texts and to recognize the oral and written text as a coherent and cohesive unit in terms of form and content. Once these objectives have been achieved, the student will understand that oral and written communication skills are central competences of his / her university life and, later, his / her professional life.

(b) **Prerrequisites:** None

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- With this course the student develops and strengthens his oral and written communication skills in the context of an academic context. In addition, it comprehends conceptually and metaphorically expository texts, and identifies the objectives, hierarchy of ideas and structure of those texts. At the end of the course, the student is able to produce descriptive and informative expository texts. Likewise, it develops its capacity of openness and tolerance towards the diversity of points of view thanks to the continuous group work, self-evaluations and peer evaluations that will be faced during the course cycle.

8. Contribution to Outcomes

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Usage**)

f) An ability to communicate effectively. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

n) Apply knowledge of the humanities in their professional work. (**Usage**)

9. Competences (IEEE)

C17. Ability to properly express in oral and written media as expected from a university graduate. ⇒ **Outcome f,h,n**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome f,n**

C24. Understanding the need for lifelong learning and improving skills and abilities.⇒ **Outcome f,h**

10. List of topics

1. Communication Laboratory I

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

Unit 1: Communication Laboratory I (12)	
Competences Expected: 4	
Learning Outcomes	**Topics**
• Selt-assessment: the students are able to recognize their own strenghts and weaknesses to make constructive criticism of their own work .	• Approach to some characteristics of formal writing. • Academic writing features • Reading strategies. • Structure of text. • Structure of paragraphs. • Characteristics of the paragraph. • Argumentative vs.expository text. • Writing process: delimitation of topic and outline production. • Citations:function and types • Approach to characteristic of the oral presentacion. • Conference:formal presentation • Writing full tet with citations.
Readings : [D93]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: EG0003. Mathematics I (Mandatory)
2. Credits: 4
3. Hours of theory and Lab: 4 HT; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[Lar14] Ron Larson. *Calculus*. 10th. CENGAGE Learning, 2014.

[Ste12] James Stewart. *Calculus*. 7th. CENGAGE Learning, 2012.

6. Information about the course

(a) **Brief description about the course** The course aims to develop in students the skills to deal with models in science and engineering related to single variable differential calculus skills. In the course it is studied and applied concepts related to calculation limits, derivatives and integrals of real and vector functions of single real variables to be used as base and support for the study of new contents and subjects. Also seeks to achieve reasoning capabilities and applicability to interact with real-world problems by providing a mathematical basis for further professional development activities.

(b) **Prerrequisites:** None

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Apply the concepts of complex numbers and functions to solve problems related to science.
- Apply mathematical concepts and techniques of differential calculus of one variable to solve problematic situations of science.
- Calculate mathematical expressions of indefinite integrals with accuracy, order and clarity in the treatment of the data.

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome j**

C24. Understanding the need for lifelong learning and improving skills and abilities.⇒ **Outcome j**

10. List of topics

1. Complex numbers
2. Functions of a single variable
3. Limits and derivatives
4. Integrals

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Complex numbers (20)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Define and operates with complex numbers, calculating their the polar and exponential form [Assessment]. • Use Moivre's theorem to simplify calculations of complex [Assessment].	• Operations with complex numbers • Moivre's theorem
Readings : [Ste12], [Lar14]	

Unit 2: Functions of a single variable (10)	
Competences Expected: C20	
Learning Outcomes	**Topics**
• Define a function of single variable and understand and be able to determine its domain and range. [Assessment]. • Recognize different specific types of functions and create scatter plots and select an appropriate model. [Assessment]. • Understand how a change in base affects the graph of exponentials and logarithmic functions. [Assessment]. • Recognizes and builds trigonometric functions.[Assessment]. • Apply rules to transform functions[Assessment]. • Be able to solve simple applications problems such as regression and curve fitting. [Assessment].	• Domain and range. • Types of functions. • Graph of exponentials and logarithmic functions. • Trigonometric functions. • Apply rules to transform functions. • Applications problems using Excel,modelling bacterial growing, Logarithmic scale, etc.
Readings : [Ste12], [Lar14]	

Unit 3: Limits and derivatives (20)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Understand the concept of limits and guess limits from the graph of a function. [Assessment]. • Find limits using the limit laws and algebraic simplification. [Assessment]. • Find vertical and horizontal asymptotes. [Assessment]. • Compute and estimate derivatives. [Assessment]. • Interpret the derivative as a rate of change. [Assessment]. • Find the derivatives of basic and composed function[Assessment]. • Approximates functions using derivate concepts and compute relative errors. [Assessment]. • Find critical numbers, and absolute and local maximum and minimum values for continuous function. [Assessment]. • Apply L'Hospital theorem to calculate some limits. [Assessment]. • Understand optimization problems, find the function to be optimized and solve.[Assessment]. • Be able to solve simple applications problems. [Assessment].	• Limits • Derivatives • Derivate concepts and compute relative errors. • L'Hospital theorem • Applications problems such as velocity, exponential growth and decay, pile increasing gravel, optimization of a can, etc
Readings : [Ste12], [Lar14]	

Unit 4: Integrals (22)	
Competences Expected: C20	
Learning Outcomes	**Topics**
• Solve properly estimate area using left and right endpoint and midpoint rectangles. [Assessment]. • Use the Fundamental theorem to find derivatives of functions of evaluate definite and indefinite integrals using substitution. [Assessment]. • Use different technic to integrate functions. [Assessment]. • Apply integrals to found areas. [Assessment]. • Compute volumes of solids obtained by rotating a bounded region about either the x-axis or the y-axis. [Assessment]. • Compute the volume of solids obtained by rotating a bounded region about either the x-axis or the y-axis by considering cylindrical shells. [Assessment]. • Compute the average value of a function. [Assessment]. • Compute work done by a force and compute center of mass for a flat plate in the plane. [Assessment]. • Define parametric curves and vectorials functions finding relationships between them. [Assessment]. • Apply integrals to calculate the length of curves described by a vectorial functions[Assessment]. • Be able to solve simple applications problems such as traffic on an Internet service, fuel consumption, tomography: volume of the brain, pump the water, mass in thickener, superformula, volume in Wankel machine, length of DNA molecule helix, etc. [Assessment].	• Strategy for integration. • Technic to integrate functions. • Additional Tools to Find Integrals. • Applications problems.
Readings : [Ste12], [Lar14]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: EG0004. Global Challenges (Mandatory)
2. Credits: 3
3. Hours of theory and Lab: 2 HT; 2 HP; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[E15] Upton. E. *Intuición, acción, creación: Graphic Design Thinking.* México:Editorial Gustavo Gili, 2015.

[R12] Curedale. R. *Design methods 1: 200 ways to apply design thinking.* EE.UU Design Community College Inc, 2012.

6. Information about the course

(a) **Brief description about the course** During the plenary sessions, there will be lectures related to the methodology of Design Thinking as well as its use and importance in the creation processes. Also, during these sessions we will have presentations on entrepreneurships and startups related to engineering or technology. During lab sessions, students form teams that maintain during the cycle. With the guidance of the teacher and through the methodology of Design Thinking developed in the plenary sessions, students will have to present innovative solutions to real problems inspired by the United Nations ”Global Challenges”. The students will have a Digital Log which will be constantly reviewed by the teachers in charge. In it will be the advances, processes and referents of the group project. The course culminates with the presentations of the proposals put forward by the groups.

(b) **Prerrequisites:** None

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Ability to design and carry out experiments.
- Ability to analyze information.
- Ability to design a system, a component or a process to meet the desired needs within realistic constraints (Level 1)
- Teamwork Ability.
- Ability to lead a team.
- Oral communication skills (Level 1)
- Written communication skills (Level 1)
- Understand the impact of engineering solutions in a global, economic, environmental and societal context.

8. Contribution to Outcomes

n) Apply knowledge of the humanities in their professional work. (**Usage**)

ñ) Understand that the formation of a good professional is not disconnected or opposed but rather contributes to genuine personal growth. This requires the assimilation of solid values, broad spiritual horizons and a deep vision of the cultural environment. (**Usage**)

9. Competences (IEEE)

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome n,ñ**

10. List of topics

1. Global Challenges.

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Global Challenges. (12)	
Competences Expected: 4	
Learning Outcomes	**Topics**
• Flexibility and Adaptability: Students learn to work in a team in a flexible and variable environment with constant challenges.	• Methodology of Design Thinking (DT). • DT Steps. • Technique and use of Brainstorming. • Knowledge of the user, empathy and use of archetypes. • Types of research, differences and uses. • Strategies for gathering from Insights. • Ideation methods. • Introduction tool Prototyping. • Introducction to User Experience. • Testing and Iteration Strategies. • Uses of Storytelling.
Readings : [R12], [E15]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS1102. Objects oriented programming I (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[PH13] Deitel. P.J and Deitel. H.M. *C++ How to Program (Early Objects Version)*. Deitel, How to Program. Prentice Hall, 2013. ISBN: 9780133378719. URL: `http://books.google.com.pe/books?id=XIZJNQEACAAJ`.

[Str13] Bjarne Stroustrup. *The C++ Programming Language*. 4th. Addison-Wesley, 2013. ISBN: 978-0-321-56384-2.

6. **Information about the course**

(a) **Brief description about the course** This is the second course in the sequence of introductory courses in computer science. The course will introduce students in the various topics of the area of computing such as: Algorithms, Data Structures, Software Engineering, etc.

(b) **Prerrequisites:** CS1100. Introduction to Computer Science. (1^{st} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Introduce the student to the foundations of the object orientation paradigm, allowing the assimilation of concepts necessary to develop information systems.

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

9. **Competences (IEEE)**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C5. Ability to implement algorithms and data structures in software.⇒ **Outcome b**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome b**

CS3. Analyze the extent to which a computer-based system meets the criteria defined for its current use and future development.⇒ **Outcome i**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C5. Ability to implement algorithms and data structures in software.⇒ **Outcome b**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome b**

CS3. Analyze the extent to which a computer-based system meets the criteria defined for its current use and future development.⇒ **Outcome i**

10. List of topics

1. General overwiew of Programming Languages
2. Virtual Machines
3. Basic Type Systems
4. Fundamental Programming Concepts
5. Object-Oriented Programming
6. Algorithms and Design
7. Algorithmic Strategies
8. Basic Analysis
9. Fundamental Data Structures and Algorithms

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: General overwiew of Programming Languages (1)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Discuss the historical context for several programming language paradigms [Familiarity]	• Brief review of programming paradigms. • Comparison between functional programming and imperative programming. • History of programming languages.
Readings : [Str13], [PH13]	

Unit 2: Virtual Machines (1)	
Competences Expected: C2, CS6	
Learning Outcomes	**Topics**
• Explain the concept of virtual memory and how it is realized in hardware and software [Familiarity] • Differentiate emulation and isolation [Familiarity] • Evaluate virtualization trade-offs [Assessment]	• The virtual machine concept. • Types of virtualization (including Hardware/Software, OS, Server, Service, Network). • Intermediate languages.
Readings : [Str13], [PH13]	

Unit 3: Basic Type Systems (2)	
Competences Expected: C1,C2,CS1	
Learning Outcomes	**Topics**
• For both a primitive and a compound type, informally describe the values that have that type [Familiarity] • For a language with a static type system, describe the operations that are forbidden statically, such as passing the wrong type of value to a function or method [Familiarity] • Describe examples of program errors detected by a type system [Familiarity] • For multiple programming languages, identify program properties checked statically and program properties checked dynamically [Usage] • Give an example program that does not type-check in a particular language and yet would have no error if run [Familiarity] • Use types and type-error messages to write and debug programs [Usage] • Explain how typing rules define the set of operations that are legal for a type [Familiarity] • Write down the type rules governing the use of a particular compound type [Usage] • Explain why undecidability requires type systems to conservatively approximate program behavior [Familiarity] • Define and use program pieces (such as functions, classes, methods) that use generic types, including for collections [Usage] • Discuss the differences among generics, subtyping, and overloading [Familiarity] • Explain multiple benefits and limitations of static typing in writing, maintaining, and debugging software [Familiarity]	• A type as a set of values together with a set of operations – Primitive types (e.g., numbers, Booleans) – Compound types built from other types (e.g., records, unions, arrays, lists, functions, references) • Model statement (link, visibility, scope and life time). • General view of type checking.
Readings : [Str13], [PH13]	

<table>
<tr><td colspan="2"><b>Unit 4: Fundamental Programming Concepts (6)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: C1,C2,CS2</b></td></tr>
<tr><td><b>Learning Outcomes</b></td><td><b>Topics</b></td></tr>
<tr><td>
<ul>
<li>Analyze and explain the behavior of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion. [Assessment]</li>
<li>Identify and describe uses of primitive data types [Familiarity]</li>
<li>Write programs that use primitive data types [Usage]</li>
<li>Modify and expand short programs that use standard conditional and iterative control structures and functions [Usage]</li>
<li>Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing [Usage]</li>
<li>Write a program that uses file I/O to provide persistence across multiple executions [Usage]</li>
<li>Choose appropriate conditional and iteration constructs for a given programming task [Assessment]</li>
<li>Describe the concept of recursion and give examples of its use [Familiarity]</li>
<li>Identify the base case and the general case of a recursively-defined problem [Assessment]</li>
</ul>
</td><td>
<ul>
<li>Basic syntax and semantics of a higher-level language</li>
<li>Variables and primitive data types (e.g., numbers, characters, Booleans)</li>
<li>Expressions and assingments</li>
<li>Simple I/O including file I/O</li>
<li>Conditional and iterative control structures</li>
<li>Functions and parameter passing</li>
</ul>
</td></tr>
<tr><td colspan="2"><b>Readings :</b> [Str13], [PH13]</td></tr>
</table>

Unit 5: Object-Oriented Programming (10)	
Competences Expected: C2,C24,CS1,CS2	
Learning Outcomes	**Topics**
• Design and implement a class [Usage] • Use subclassing to design simple class hierarchies that allow code to be reused for distinct subclasses [Usage] • Correctly reason about control flow in a program using dynamic dispatch [Usage] • Compare and contrast (1) the procedural/functional approach—defining a function for each operation with the function body providing a case for each data variant—and (2) the object-oriented approach—defining a class for each data variant with the class definition providing a method for each operation Understand both as defining a matrix of operations and variants [Assessment] • Explain the relationship between object-oriented inheritance (code-sharing and overriding) and subtyping (the idea of a subtype being usable in a context that expects the supertype) [Familiarity] • Use object-oriented encapsulation mechanisms such as interfaces and private members [Usage] • Define and use iterators and other operations on aggregates, including operations that take functions as arguments, in multiple programming languages, selecting the most natural idioms for each language [Usage]	• Object-oriented design – Decomposition into objects carrying state and having behavior – Class-hierarchy design for modeling • Object-oriented idioms for encapsulation – Privacy and visibility of class members – Interfaces revealing only method signatures – Abstract base classes • Definition of classes: fields, methods, and constructors • Subclasses, inheritance, and method overriding • Subtyping – Subtype polymorphism; implicit upcasts in typed languages – Notion of behavioral replacement: subtypes acting like supertypes – Relationship between subtyping and inheritance • Using collection classes, iterators, and other common library components • Dynamic dispatch: definition of method-call
Readings : [Str13], [PH13]	

Unit 6: Algorithms and Design (3)	
Competences Expected: C1,C2,C23,CS6	
Learning Outcomes	**Topics**
• Discuss the importance of algorithms in the problem-solving process [Familiarity] • Discuss how a problem may be solved by multiple algorithms, each with different properties [Familiarity] • Create algorithms for solving simple problems [Usage] • Use a programming language to implement, test, and debug algorithms for solving simple problems [Usage] • Implement, test, and debug simple recursive functions and procedures [Usage] • Determine whether a recursive or iterative solution is most appropriate for a problem [Assessment] • Implement a divide-and-conquer algorithm for solving a problem [Usage] • Apply the techniques of decomposition to break a program into smaller pieces [Usage] • Identify the data components and behaviors of multiple abstract data types [Usage] • Implement a coherent abstract data type, with loose coupling between components and behaviors [Usage] • Identify the relative strengths and weaknesses among multiple designs or implementations for a problem [Assessment]	• Problem-solving strategies – Iterative and recursive mathematical functions – Iterative and recursive traversal of data structures – Divide-and-conquer strategies • The role of algorithms in the problem-solving process • Problem-solving strategies – Iterative and recursive mathematical functions – Iterative and recursive traversal of data structures – Divide-and-conquer strategies • Fundamental design concepts and principles – Abstraction – Program decomposition – Encapsulation and information hiding – Separation of behaivor and implementation
Readings : [Str13], [PH13]	

Unit 7: Algorithmic Strategies (3)	
Competences Expected: C1,C2,C24,CS1	
Learning Outcomes	**Topics**
• For each of the strategies (brute-force, greedy, divide-and-conquer, recursive backtracking, and dynamic programming), identify a practical example to which it would apply [Familiarity] • Use a greedy approach to solve an appropriate problem and determine if the greedy rule chosen leads to an optimal solution [Assessment] • Use a divide-and-conquer algorithm to solve an appropriate problem [Usage] • Use recursive backtracking to solve a problem such as navigating a maze [Usage] • Use dynamic programming to solve an appropriate problem [Usage] • Determine an appropriate algorithmic approach to a problem [Assessment] • Describe various heuristic problem-solving methods [Familiarity]	• Brute-force algorithms • Greedy algorithms • Divide-and-conquer • Recursive backtracking • Dynamic Programming
Readings : [Str13], [PH13]	

Unit 8: Basic Analysis (2)	
Competences Expected: C1,C2,C24,CS1	
Learning Outcomes	**Topics**
• Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm [Familiarity]	• Differences among best, expected, and worst case behaviors of an algorithm
Readings : [Str13], [PH13]	

Unit 9: Fundamental Data Structures and Algorithms (6)	
Competences Expected: C1,C2,C24,CS1	
Learning Outcomes	**Topics**
• Implement basic numerical algorithms [Usage] • Implement simple search algorithms and explain the differences in their time complexities [Assessment] • Be able to implement common quadratic and O(N log N) sorting algorithms [Usage] • Discuss the runtime and memory efficiency of principal algorithms for sorting, searching, and hashing [Familiarity] • Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Familiarity] • Explain how tree balance affects the efficiency of various binary search tree operations [Familiarity] • Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context [Assessment] • Trace and/or implement a string-matching algorithm [Usage]	• Simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max, • Sequential and binary search algorithms • Worst case quadratic sorting algorithms (selection, insertion) • Worst or average case O(N log N) sorting algorithms (quicksort, heapsort, mergesort)
Readings : [Str13], [PH13]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS1D02. Discrete Structures II (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HP; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction.* 5 ed. Pearson, 2003.

[Gri97] R. Grimaldi. *Matemáticas Discretas y Combinatoria.* Addison Wesley Iberoamericana, 1997.

[Joh99] Richard Johnsonbaugh. *Matemáticas Discretas.* Prentice Hall, México, 1999.

[Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications.* 7 ed. Mc Graw Hill, 2007.

6. **Information about the course**

(a) **Brief description about the course** In order to understand the advanced computational techniques, the students must have a strong knowledge of the Various discrete structures, structures that will be implemented and used in the laboratory in the programming language..

(b) **Prerrequisites:** CS1D01. Discrete Structures I. (1^{st} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- That the student is able to model computer science problems using graphs and trees related to data structures
- That the student applies efficient travel strategies to be able to search data in an optimal way

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Familiarity**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Familiarity**)

a) An ability to apply knowledge of mathematics, science. (**Familiarity**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Familiarity**)

9. **Competences (IEEE)**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C5. Ability to implement algorithms and data structures in software.⇒ **Outcome b**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome i**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C5. Ability to implement algorithms and data structures in software.⇒ **Outcome b**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome i**

10. List of topics

1. Digital Logic and Data Representation
2. Basics of Counting
3. Graphs and Trees

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

<table>
<tr><td colspan="2"><b>Unit 1: Digital Logic and Data Representation (10)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: C1,C20</b></td></tr>
<tr><td><b>Learning Outcomes</b></td><td><b>Topics</b></td></tr>
<tr><td>• Explain the importance of Boolean algebra as a unification of set theory and propositional logic [Assessment].
• Explain the algebraic structures of reticulum and its types [Assessment].
• Explain the relationship between the reticulum and the ordinate set and the wise use to show that a set is a reticulum [Assessment].
• Explain the properties that satisfies a Boolean algebra [Assessment].
• Demonstrate if a terna formed by a set and two internal operations is or not Boolean algebra [Assessment].
• Find the canonical forms of a Boolean function [Assessment].
• Represent a Boolean function as a Boolean circuit using logic gates [Assessment].
• Minimize a Boolean function. [Assessment].</td><td>• Reticles: Types and properties.
• Boolean algebras.
• Boolean Functions and Expressions.
• Representation of Boolean Functions: Normal Disjunctive and Conjunctive Form.
• Logical gates.
• Circuit Minimization.</td></tr>
<tr><td colspan="2"><b>Readings :</b> [Ros07], [Gri03]</td></tr>
</table>

Unit 2: Basics of Counting (40)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Apply counting arguments, including sum and product rules, inclusion-exclusion principle and arithmetic/geometric progressions [Familiarity] • Apply the pigeonhole principle in the context of a formal proof [Familiarity] • Compute permutations and combinations of a set, and interpret the meaning in the context of the particular application [Familiarity] • Map real-world applications to appropriate counting formalisms, such as determining the number of ways to arrange people around a table, subject to constraints on the seating arrangement, or the number of ways to determine certain hands in cards (eg, a full house) [Familiarity] • Solve a variety of basic recurrence relations [Familiarity] • Analyze a problem to determine underlying recurrence relations [Familiarity] • Perform computations involving modular arithmetic [Familiarity]	• Counting arguments – Set cardinality and counting – Sum and product rule – Inclusion-exclusion principle – Arithmetic and geometric progressions • The pigeonhole principle • Permutations and combinations – Basic definitions – Pascal's identity – The binomial theorem • Solving recurrence relations – An example of a simple recurrence relation, such as Fibonacci numbers – Other examples, showing a variety of solutions • Basic modular arithmetic
Readings : [Gri97]	

Unit 3: Graphs and Trees (40)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Illustrate by example the basic terminology of graph theory, and some of the properties and special cases of each type of graph/tree [Familiarity] • Demonstrate different traversal methods for trees and graphs, including pre, post, and in-order traversal of trees [Familiarity] • Model a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system [Familiarity] • Show how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting [Familiarity] • Explain how to construct a spanning tree of a graph [Familiarity] • Determine if two graphs are isomorphic [Familiarity]	• Trees – Properties – Traversal strategies • Undirected graphs • Directed graphs • Weighted graphs • Spanning trees/forests • Graph isomorphism
Readings : [Joh99]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** ME0019. Physics I (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 4 HT; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. Bibliography

[AF95] Marcelo Alonso and Edward Finn. *Física*. Addison Wesley Iberoamericana, 1995. ISBN: 0-201-62565-2.

[SB02] Raymond Serway and Robert Beichner. *Física, para Ciencias e Ingenierias*. Mc Graw Hill, 2002. ISBN: 970-10-3581-X.

6. Information about the course

(a) **Brief description about the course** This course is useful in this career so that the student learns to show a high degree of mastery of the laws of the movement of General Physics.

(b) **Prerrequisites:** None

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Train and present to the student the basic principles of Physics as a natural science encompassing its most important topics and their relation with everyday problems.

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome i,j**

10. List of topics

1. FI1. Introducción
2. FI2. Movimiento de partículas en una dimensión
3. FI3. Movimiento de partículas en dos y tres dimensiones

4. FI4. Leyes del movimiento
5. FI5. Trabajo y Energía
6. FI6. Momento lineal
7. FI7. Rotación de cuerpos rígidos
8. FI8. Dinámica del movimiento de rotación

11. Methodology and Evaluation

Methodology:

Theory Sessions:

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:

In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:

Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: FI1. Introducción (4)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Understand and work with the physical magnitudes of the SI. • Abstracting the rigorous physical concepts of nature and represent them in vector models. • Understand and apply vector concepts to real physical problems.	• Scientific research. The cientific method. • Concept of Chemistry. Chemistry today • Matter.Classification and physical, chemical, intensive and extensive properties. • Idealized model. • Physical magnitudes. • Properties of vectors • Components of a vector and unit vectors. • Vector product. • Exercises and problems.
Readings : [SB02], [AF95]	

Unit 2: FI2. Movimiento de partículas en una dimensión (2)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Describe mathematically the mechanical motion of a one-dimensional particle as a body of negligible dimensions • Know and apply concepts of kinematic magnitudes. • Describe the particle motion behavior, theoretically and graphically • Knowing one-dimensional vector representations of these movements. • Solve problems.	• Displacement,Velocity,Speed. • Instant velocity • Medium and Instant Acceleration. • Movement with constant acceleration • Free fall of bodies • Exercises and problems.
Readings : [SB02], [AF95]	

Unit 3: FI3. Movimiento de partículas en dos y tres dimensiones (4)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Describe mathematically the mechanical motion of a particle in two and three dimensions as a body of negligible dimensions. • Know and apply concepts of vector kinematic quantities in two and three dimensions. • Describe the behavior of particle motion theoretically and graphically in two and three dimensions • Know and apply circular movement concepts. • Solve problems.	• Displacement and Velocity. • The vector acceleration • Parabolic movement. • Circular movement • Tangential and radial acceleration components. • Exercises and problems.
Readings : [SB02], [AF95]	

Unit 4: FI4. Leyes del movimiento (6)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Know the concepts of force. • know the most important interactions of nature and to represent them in a free-body diagram • Know the concepts of static equilibrium. • know and apply the laws of motion and to characterize them vectorially. • Know and apply Newton's laws. • Solve problems.	• Force and interactions. • Newton's First Law • Inertial mass. • Newton's Second Law • Weight. • Free Body Diagrams. • Newton's Third Law • Friction forces. • Dynamics of circular motion • Exercises and problems.
Readings : [SB02], [AF95]	

Unit 5: FI5. Trabajo y Energía (4)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Establish the concepts of physical energy. (Classical Physics) • Know some forms of energy. • Establish the relation between work and energy • Know and apply the concepts of energy conservation • Solve problems.	• Trabajo realizado por una fuerza constante. • Trabajo realizado por fuerzas variables. • Work and kinetic energy. • Potency • Gravitational potential energy • Elastic potential energy • Conservative and non-conservative forces • Principles of energy conservation • Exercises and problems
Readings : [SB02], [AF95]	

Unit 6: FI6. Momento lineal (3)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Establish the concepts of linear momentum. • Know the concepts of conservation of linear momentum • Know the movement of a system of particles • Solve problems.	• Linear momentum. • Conservation of linear momentum • Mass and gravity center • Movement of a particle system • Exercises and problems.
Readings : [SB02], [AF95]	

Unit 7: FI7. Rotación de cuerpos rígidos (4)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Know the basic concepts of rigid body. • Know and apply concepts of rigid body rotation. • Know torsion concepts. • Apply energy concepts to the rotating motion. • Solve problems.	• Velocity and angular accelerations. • Rotation with constant angular acceleration. • Relation between linear and angular kinematics • Energía en el movimiento de rotación. • Torsional moment. • Relationship between torsional moment and angular acceleration. • Exercises and problems.
Readings : [SB02], [AF95]	

Unit 8: FI8. Dinámica del movimiento de rotación (3)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Understand basic concepts of rotation dynamics. • Know and apply torsion concepts. • Understanding angular momentum and its conservation • Solve problems.	• Moment of torsion and angular acceleration of a rigid body • Rotation of a rigid body on a movable axis. • Work and potency in the rotation movement • Angular momentum. • Conservation of angular momentum • Exercises and problems.
Readings : [SB02], [AF95]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** GH0006. Communication Laboratory II (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 2 HT; 2 HP; (15 weeks)
4. **Professor(s)**

   Meetings after coordination with the professor

5. **Bibliography**

[D08] Cassany. D. *Prácticas letradas contemporáneas.* DF,México,Ríos de tinta, 2008.

6. **Information about the course**

   (a) **Brief description about the course** This laboratory is oriented to consolidate the student's communicative skills, both oral and written in the framework of the discipline under study. In particular, the student will strengthen his / her expositive abilities by exercising throughout the first part of the course in writing a type of text that will develop throughout his career as an engineer: laboratory reports. He will reflect on the rhetorical situation he faces when writing this type of text: who will be his reader, what is the communicative intention of that text and the subject on which he is writing. In a second part, the course is presented as a space for discussion about argumentative discourse and critical reading of argumentative texts, so that the student reflects, knows and uses the communicative tools to produce formal argumentative texts. In this sense, the course is oriented towards the production Permanent written and oral texts, so that the student will participate not only in discussion forums but is expected to be able to discuss with his colleagues on a topic proposed by the teacher. In short, the course seeks to consolidate the skills of reading, analysis and preparation of written and oral texts, both expository and argumentative.

   (b) **Prerrequisites:** GH0005. Communication Laboratory I. (1^{st} Sem)

   (c) **Type of Course:** Mandatory

   (d) **Modality:** Face to face

7. **Specific goals of the Course**

   - Develop skills that enable students to improve their communication skills, both oral and written.
   - Understand and produce expository texts in which they report on the application of theoretical knowledge in a different experiment or context.
   - Understand and produce oral and written argumentative texts.
   - Be able to discuss using solid arguments.
   - Use appropriately and reflexively the information obtained from different sources.
   - Show openness and respect to listen to the diversity of opinions or points of view of classmates

8. **Contribution to Outcomes**

   i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

   f) An ability to communicate effectively. (**Usage**)

9. **Competences (IEEE)**

C24. Understanding the need for lifelong learning and improving skills and abilities.⇒ **Outcome n,ñ**

10. List of topics

1. Communication Laboratory II

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Communication Laboratory II (12)	
Competences Expected: C24	
Learning Outcomes	**Topics**
• Appropriately handle the system of citation and bibliographical references,and recognize the importance of it is use.	• What is lab report ? • Laboratory development or methodology. • Laboratory results and apploications. • Introduction and conclusions. • Quotation, parenthetical references and bibliography construction. • Review characteristics of orality. • Presentation of an argumentative text:formal text and informal texts. • How do you build an argument? • Pragmatic argument. • Quotation,references and APA format. • Counter argumentation
Readings : [D08]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: GH0007. Introduction to Business Development (Mandatory)
2. Credits: 2
3. Hours of theory and Lab: 1 HT; 2 HP; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[AY10] Osterwalder. A and Pigneur. Y. *Business Model Generation.* ., 2010.

6. Information about the course

(a) **Brief description about the course** This course aims to provide students with a real-life hands-on experience in the first steps within a business life cycle, through which an idea becomes a formal business model. It is the first of a set of three courses designed to accompany students as they transform an idea into a prospective business or business, from idea to review of current business strategy.

(b) **Prerrequisites:** GH0005. Communication Laboratory I. (1^{st} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Ability to analyze information.
- Interpretation of information and results.
- Teamwork Ability.
- Ethics.
- Oral communication.
- Written communication.
- Graphic communication.
- Understand the need to learn continuously

8. Contribution to Outcomes

d) An ability to function on multidisciplinary teams. (**Usage**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Usage**)

f) An ability to communicate effectively. (**Usage**)

n) Apply knowledge of the humanities in their professional work. (**Usage**)

o) Improve the conditions of society by putting technology at the service of the human being. (**Usage**)

9. Competences (IEEE)

C10. Understanding of the impact on individuals, organizations, and society of deploying technological solutions and interventions.⇒ **Outcome d,n,o**

C17. Ability to properly express in oral and written media as expected from a university graduate. ⇒ **Outcome f**

C18. Ability to participate actively and as a member of a team. .⇒ **Outcome e**

10. List of topics

1. Introduction to Business Development.

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Introduction to Business Development. (12)	
Competences Expected: 4	
Learning Outcomes	**Topics**
• Transform an initial business idea or an innovation process into a feasible business model.	• The business lifecycle: from the idea to reviewing your strategy. • The ideation process and the customer's view. • How to build and maintain effective teams? • Running LEAN: the basics. • Designing a business model: design tools and Canvas. • Business Model generation: the Canvas Business Model (Osterwalder). • Venture Engineering: using computer science skills to build an effective business model. • Primary market research tools and market niches. • The Importance of Capital: Human, Financial and Intellectual. • Monetizing and Financing Techniques. • Effective communication: making an impact business model presentation.
Readings : [AY10]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** GH1002. Art and Technology (Mandatory)
2. **Credits:** 1
3. **Hours of theory and Lab:** 2 HP; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[J12] Maeda J. *Processing: A Programming Handbook for Visual Designers and Artists.* Cambridge: The MIT Press, 2012.

[S02] Wilson. S. *Intersections of Art, Science and Technology.* Cambridge: The MIT Press, 2002.

6. **Information about the course**

(a) **Brief description about the course** The course seeks to give a global, historical and critical vision of the transformations and synergies of contemporary art. Where students approach two components of contemporary art and design: interdisciplinary practices and points of contact between the arts and the technological and engineering processes.

(b) **Prerrequisites:** EG0004. Global Challenges. (1st Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Develop the ability to analyze information.
- Develop the ability to interpret information.
- Develop the ability to work as a team.
- Developing Oral communication skills.
- Recognize the need for lifelong learning.

8. **Contribution to Outcomes**

d) An ability to function on multidisciplinary teams. (**Usage**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Usage**)

f) An ability to communicate effectively. (**Usage**)

n) Apply knowledge of the humanities in their professional work. (**Usage**)

o) Improve the conditions of society by putting technology at the service of the human being. (**Usage**)

9. **Competences (IEEE)**

C10. Understanding of the impact on individuals, organizations, and society of deploying technological solutions and interventions.⇒ **Outcome d,n,o**

C17. Ability to properly express in oral and written media as expected from a university graduate. ⇒ **Outcome f**

C18. Ability to participate actively and as a member of a team. .⇒ **Outcome f**

C21. Understanding the professional, legal, security, political, humanistic, environmental, cultural and ethical issues. ⇒ **Outcome e**

10. List of topics

1. Arts and Technology.
2. Digital Art
3. Prototyping, analysis and creation

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Arts and Technology. (12)	
Competences Expected: 4	
Learning Outcomes	**Topics**
• Promote the interest in learn about current issues of Peruvian society and the world.	• What is art and what is it for? • The artistic discourse: identity, territory, politics and society.
Readings : [S02]	

Unit 2: Digital Art (24)	
Competences Expected: 3	
Learning Outcomes	**Topics**
• Development of skills such as: creativity, critical thinking, observation and synthesis.	• Generative Art. • Net Art. • Virtual Reality.
Readings : [J12]	

Unit 3: Prototyping, analysis and creation (24)	
Competences Expected: 3	
Learning Outcomes	**Topics**
• Students understand the importance and effectiveness of teamwork, in both academic and professional life. During the semester, students perform group and individual activities whose common goal is the generation of a project that links concepts of art, technology and engineering.	• Digital Manufacturing • Intervention: Action and public space • Presentation: Montage and portafolio
Readings : [S02]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** GH1101. English I (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 10 HP; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Cam06] Cambridge. *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford, 2006.

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

6. **Information about the course**

(a) **Brief description about the course** A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of people's lives. In the case of foreign languages, English is undoubtedly the most pratical because it is spoken around the world. There is no country where it is not spoken. In careers related to tourist services English is perhaps the most important practical tool that the student must master from the outset as part of his comprehensive education.

(b) **Prerrequisites:** None

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Know the English language and its grammatical structure.
- Identify situations and employ dialogues related to them.

8. **Contribution to Outcomes**

f) An ability to communicate effectively. (**Usage**)

9. **Competences (IEEE)**

C25. Ability to communicate in a second language.⇒ **Outcome f**

10. **List of topics**

1. Hello everybody!
2. Meeting people!
3. The world of work
4. Take it easy!
5. Where do you live?

6. Can you speak English?
7. Then and now!

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Hello everybody! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the first unit, each student, understanding the grammar of the present tense is able to express a greater quantity of expressions of time and also to use sentences with the verb To Be to express situation and state. • That the student is able to analyze and express ideas about dates and numbers in order.	• Verb To Be. • Affirmative sentences , Negatives and Questions. • Number Expressions. • Objects and Countries. • Expressions to greet and make presentations.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 2: Meeting people! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the second unit, students having identified how to express orders and make offerings in restaurants use them in various situations. Explain and apply food and drink vocabulary.	• Possessive adjectives. • Expressions to find prices. • Possession expressions • Vocabulary of Family, Food and Drinks. • Formal requests. • Informal letters.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 3: The world of work (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the third unit, students having recognized the characteristics of the present simple, use it to make descriptions of various types. Describe people and places and give directions. Express time.	• Simple present tense. Auxiliaries. • Affirmative sentences, Negatives and Questions. • Common verbs and occupations. • Indications for expressing the time.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 4: Take it easy! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the conclusion of the fourth unit, the students having identified the idea of expressing ideas of free time actions in Simple and Continuous Present. Express ideas of stations and related activities.	• Present Simple 2. • Affirmative sentences , Negatives and Questions. • Use of Verbs of entertainment. • Free time. • The seasons of the year. • Expressions of social activities.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 5: Where do you live? (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the fifth unit, students, based on the understanding of the present continuous time, will elaborate sentences using ideas of location and place. They will also assimilate the need to express objects in common use. They will acquire vocabulary to describe the parts of a house using expressions to ask for directions.	• Use There is/There are. • Sentences with Prepositions. • Expressions of Quantity. • Vocabulary of airplanes and places. • Expressions of direction indications.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 6: Can you speak English? (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the sixth unit, students having learned the fundamentals of using auxiliary mode, will create sentences applied to the appropriate context. They emphasize the difference between languages and nationalities. They describe feelings. Use expressions on the phone.	• Can/cant. • Past of verb to be. Use of Could • Vocabulary of Countries and languages. • Expressions for using the phone • Writing formal letters. • Readings.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 7: Then and now! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the seventh unit, students having learned the basics of structuring the Simple Past experience the need to be able to express this type of time in actions. They will practice in appropriate contexts. They emphasize the difference between irregular and regular verbs. They describe actions with several verbs. They use expressions to describe the climate.	• Past Simple. • Expressions of past tense. • Vocabulary regular and irregular verbs • Expressions to describe the climate. • Writing descriptive paragraphs. • Special occasions.
Readings : [SJ02], [Cam06], [Mac99]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** EG0005. Math II (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 4 HT; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Ste12] James Stewart. *Calculus*. 7th. CENGAGE Learning, 2012.

[Zil13] Dennis G. Zill. *Differential equations with Boundary value problems*. 8th. CENGAGE Learning, 2013.

6. Information about the course

(a) **Brief description about the course** The course develops in students the skills to deal with models of science and engineering skills. In the first part of the course a study of the functions of several variables, partial derivatives, multiple integrals and an introduction to vector fields is performed. Then the student will use the basic concepts of calculus to model and solve ordinary differential equations using techniques such as Laplace transforms and Fourier series.

(b) **Prerrequisites:** EG0003. Mathematics I. (1^{st} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Apply derivation rules and partial differentation in functions of several variables.
- Apply techniques for calculating multiple integrals.
- Understand and use the concepts of vector calculus.
- Understand the importance of series.
- Identify and solve differential equations of the first order and their applications in chemical and physical problems.

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome j**

10. List of topics

1. Multi-Variable Function Differential
2. Multi-Variable function Integral
3. Series
4. Ordinary Differential Equations

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Multi-Variable Function Differential (24)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Understand the concept of multi-variable functions. • Master the concept and calculation method of the direction derivative and gradient of the guide. • Master the calculation method of the first order and second order partial derivative of composite functions. • Master the calculation method of the partial derivatives for implicit functions. • Understand tangent line, normal plane to curve line and tangent plane, normal line to a curve plan. Know to calculate their equations. • Learn the concept of extreme value and conditional extreme value of multi-variable functions; know to find out the binary function extreme value. • Be able to solve simple applications problems.	• Concept of multi-variable functions. • Directional Derivates • Tangent line, normal plane to curve line and tangent plane, normal line to a curve plan. Know to calculate their equations. • Concept of extreme value and conditional extreme value of multi-variable functions • Applications problems such as modeling total production of an economic system, speed of sound through the ocean, thickener optimization, etc.
Readings : [Ste12], [Zil13]	

Unit 2: Multi-Variable function Integral (12)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Understand the double integral, triple integral, and understand the nature of the multiple integral. • Master the calculation method of double integral (Cartesian coordinates, polar coordinates) the triple integral (Cartesian coordinates, cylindrical coordinates, spherical coordinates). • Understand the concept of line Integral, their properties and relationships. • Know to calculate the line integral. • Master the calculation the rotational, divergence and Laplacian.	• Double integral, triple integral and nature of the multiple integral. • Method of double integral • Line Integral • The Divergence, Rotation and Laplacian
Readings : [Ste12], [Zil13]	

Unit 3: Series (24)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Master to calculation if series is convergent, and if convergent, find the sum of the series trying to find the radius of convergence and the interval of convergence of a power series. • Represent a function as a power series and find the Taylor and McLaurin Series to estimate function values to a desired accuracy. • Understand the concepts of orthogonal functions and the expansion of a given function f to find its Fourier series.	• Convergent series • Taylor and McLaurin series • Orthogonal functions
Readings : [Ste12], [Zil13]	

Unit 4: Ordinary Differential Equations (30)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Understand differential equations, solutions, order, general solution, initial conditions and special solutions etc. • Master the calculation method for variables separable equation and first order linear equations. Known to solve homogeneous equation and Bernoulli (Bernoulli) equations; understand variable substitution to solve the equation. • Master to solve total differential equations. • Be able to use reduced order method to solve equations. • Understand the structure of the second order linear differential equation. • Master calculation method for the constant coefficient homogeneous linear differential equations; and understand calculation method for the higher order homogeneous linear differential equations. • Know to apply the differential equation calculation method to solve simple geometric and physic application problems. • Solve properly certain types of differential equations using Laplace transforms.	• Concept of differential equations • Methods to resolve differential equations • Methods to resolve the secod order linear differential equations • Higher order linear ordinary differential equations • Applications problems using Laplace transforms
Readings : [Ste12], [Zil13]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS1103. Objects oriented programming II (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 3 HT; 2 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. Bibliography

[Nak13] S. Nakariakov. *The Boost C++ Libraries: Generic Programming.* CreateSpace Independent Publishing Platforml, 2013.

6. Information about the course

(a) **Brief description about the course** This is the third course in the sequence of introductory courses in computer science. This course is intended to cover Concepts indicated by the Computing Curriculum IEEE (c) -ACM 2001, under the functional-first approach. The object-oriented paradigm allows us to combat complexity by making models from abstractions of the problem elements and using techniques such as encapsulation, modularity, polymorphism and inheritance. The Dominion of these topics will enable participants to provide computational solutions to design problems simple of the real world.

(b) **Prerrequisites:** CS1102. Objects oriented programming I. (2^{nd} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Introduce the student in the fundaments of the paradigm of object orientation, allowing the assimilation of concepts necessary to develop an information system

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C3. An intellectual understanding of, and an appreciation for, the central role of algorithms and data structures.⇒ **Outcome a**

C18. Ability to participate actively and as a member of a team. .⇒ **Outcome d**

CS1. Model and design computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.⇒ **Outcome a**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome b**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C3. An intellectual understanding of, and an appreciation for, the central role of algorithms and data structures.⇒ **Outcome a**

C18. Ability to participate actively and as a member of a team. .⇒ **Outcome d**

CS1. Model and design computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.⇒ **Outcome a**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome b**

10. List of topics

1. Fundamental Programming Concepts
2. Algorithms and Design
3. Event-Driven and Reactive Programming
4. Basic Analysis
5. Fundamental Data Structures and Algorithms
6. Basic Type Systems
7. Object-Oriented Programming
8. Graphs and Trees
9. Software Design
10. Requirements Engineering

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Fundamental Programming Concepts (5)	
Competences Expected: C1,C18	
Learning Outcomes	**Topics**
• Analyze and explain the behavior of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion. [Usage] • Identify and describe uses of primitive data types [Usage] • Write programs that use primitive data types [Usage] • Modify and expand short programs that use standard conditional and iterative control structures and functions [Usage] • Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing [Usage] • Write a program that uses file I/O to provide persistence across multiple executions [Usage] • Choose appropriate conditional and iteration constructs for a given programming task [Usage] • Describe the concept of recursion and give examples of its use [Usage] • Identify the base case and the general case of a recursively-defined problem [Usage]	• Basic syntax and semantics of a higher-level language • Variables and primitive data types (e.g., numbers, characters, Booleans) • Expressions and assingments • Simple I/O including file I/O • Conditional and iterative control structures • Functions and parameter passing • The concept of recursion
Readings : [**stroustrup2013**]	

Unit 2: Algorithms and Design (5)	
Competences Expected: C3,C18	
Learning Outcomes	**Topics**
• Discuss the importance of algorithms in the problem-solving process [Usage] • Discuss how a problem may be solved by multiple algorithms, each with different properties [Usage] • Create algorithms for solving simple problems [Usage] • Use a programming language to implement, test, and debug algorithms for solving simple problems [Usage] • Implement, test, and debug simple recursive functions and procedures [Usage] • Determine whether a recursive or iterative solution is most appropriate for a problem [Usage] • Implement a divide-and-conquer algorithm for solving a problem [Usage] • Apply the techniques of decomposition to break a program into smaller pieces [Usage] • Identify the data components and behaviors of multiple abstract data types [Usage] • Implement a coherent abstract data type, with loose coupling between components and behaviors [Usage] • Identify the relative strengths and weaknesses among multiple designs or implementations for a problem [Usage]	• The concept and properties of algorithms – Informal comparison of algorithm efficiency (e.g., operation counts) • The role of algorithms in the problem-solving process • Problem-solving strategies – Iterative and recursive mathematical functions – Iterative and recursive traversal of data structures – Divide-and-conquer strategies • Fundamental design concepts and principles – Abstraction – Program decomposition – Encapsulation and information hiding – Separation of behaivor and implementation
Readings : [stroustrup2013]	

Unit 3: Event-Driven and Reactive Programming (2)	
Competences Expected: C1,C18	
Learning Outcomes	**Topics**
• Write event handlers for use in reactive systems, such as GUIs [Usage] • Explain why an event-driven programming style is natural in domains where programs react to external events [Usage] • Describe an interactive system in terms of a model, a view, and a controller [Usage]	• Events and event handlers • Canonical uses such as GUIs, mobile devices, robots, servers • Using a reactive framework – Defining event handlers/listeners – Main event loop not under event-handler-writer's control • Externally-generated events and program-generated events • Separation of model, view, and controller
Readings : [stroustrup2013]	

Unit 4: Basic Analysis (3)	
Competences Expected: CS2,C18	
Learning Outcomes	**Topics**
• Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm [Usage] • In the context of specific algorithms, identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors [Usage] • Determine informally the time and space complexity of simple algorithms [Usage] • State the formal definition of big O [Usage] • List and contrast standard complexity classes [Usage] • Perform empirical studies to validate hypotheses about runtime stemming from mathematical analysis Run algorithms on input of various sizes and compare performance [Usage] • Give examples that illustrate time-space trade-offs of algorithms [Usage] • Use big O notation formally to give asymptotic upper bounds on time and space complexity of algorithms [Usage] • Use big O notation formally to give expected case bounds on time complexity of algorithms [Usage] • Explain the use of big omega, big theta, and little o notation to describe the amount of work done by an algorithm [Usage] • Use recurrence relations to determine the time complexity of recursively defined algorithms [Usage] • Solve elementary recurrence relations, eg, using some form of a Master Theorem [Usage]	• Differences among best, expected, and worst case behaviors of an algorithm • Asymptotic analysis of upper and expected complexity bounds • Big O notation: formal definition • Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential • Empirical measurements of performance • Time and space trade-offs in algorithms • Big O notation: use • Little o, big omega and big theta notation • Recurrence relations • Analysis of iterative and recursive algorithms • Some version of a Master Theorem
Readings : [**stroustrup2013**]	

Unit 5: Fundamental Data Structures and Algorithms (3)	
Competences Expected: C3,C18	
Learning Outcomes	**Topics**
• Implement basic numerical algorithms [Usage] • Implement simple search algorithms and explain the differences in their time complexities [Usage] • Be able to implement common quadratic and O(N log N) sorting algorithms [Usage] • Describe the implementation of hash tables, including collision avoidance and resolution [Usage] • Discuss the runtime and memory efficiency of principal algorithms for sorting, searching, and hashing [Usage] • Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Usage] • Explain how tree balance affects the efficiency of various binary search tree operations [Usage] • Solve problems using fundamental graph algorithms, including depth-first and breadth-first search [Usage] • Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context [Usage] • Describe the heap property and the use of heaps as an implementation of priority queues [Usage] • Solve problems using graph algorithms, including single-source and all-pairs shortest paths, and at least one minimum spanning tree algorithm [Usage] • Trace and/or implement a string-matching algorithm [Usage]	• Simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max, • Sequential and binary search algorithms • Worst case quadratic sorting algorithms (selection, insertion) • Worst or average case O(N log N) sorting algorithms (quicksort, heapsort, mergesort) • Hash tables, including strategies for avoiding and resolving collisions • Binary search trees – Common operations on binary search trees such as select min, max, insert, delete, iterate over tree • Graphs and graph algorithms – Representations of graphs (e.g., adjacency list, adjacency matrix) – Depth- and breadth-first traversals • Heaps • Graphs and graph algorithms – Shortest-path algorithms (Dijkstra's and Floyd's algorithms) – Minimum spanning tree (Prim's and Kruskal's algorithms) • Pattern matching and string/text algorithms (e.g., substring matching, regular expression matching, longest common subsequence algorithms)
Readings : [stroustrup2013]	

Unit 6: Basic Type Systems (5)	
Competences Expected: C1,C18	
Learning Outcomes	**Topics**
• For both a primitive and a compound type, informally describe the values that have that type [Usage] • For a language with a static type system, describe the operations that are forbidden statically, such as passing the wrong type of value to a function or method [Usage] • Describe examples of program errors detected by a type system [Usage] • For multiple programming languages, identify program properties checked statically and program properties checked dynamically [Usage] • Give an example program that does not type-check in a particular language and yet would have no error if run [Usage] • Use types and type-error messages to write and debug programs [Usage] • Explain how typing rules define the set of operations that are legal for a type [Usage] • Write down the type rules governing the use of a particular compound type [Usage] • Explain why undecidability requires type systems to conservatively approximate program behavior [Usage] • Define and use program pieces (such as functions, classes, methods) that use generic types, including for collections [Usage] • Discuss the differences among generics, subtyping, and overloading [Usage] • Explain multiple benefits and limitations of static typing in writing, maintaining, and debugging software [Usage]	• A type as a set of values together with a set of operations – Primitive types (e.g., numbers, Booleans) – Compound types built from other types (e.g., records, unions, arrays, lists, functions, references) • Association of types to variables, arguments, results, and fields • Type safety and errors caused by using values inconsistently given their intended types • Goals and limitations of static typing – Eliminating some classes of errors without running the program – Undecidability means static analysis must conservatively approximate program behavior • Generic types (parametric polymorphism) – Definition – Use for generic libraries such as collections – Comparison with ad hoc polymorphism (overloading) and subtype polymorphism • Complementary benefits of static and dynamic typing – Errors early vs. errors late/avoided – Enforce invariants during code development and code maintenance vs. postpone typing decisions while prototyping and conveniently allow flexible coding patterns such as heterogeneous collections – Avoid misuse of code vs. allow more code reuse – Detect incomplete programs vs. allow incomplete programs to run
Readings : [**stroustrup2013**]	

<table>
<tr><th colspan="2"><b>Unit 7: Object-Oriented Programming (7)</b></th></tr>
<tr><th colspan="2"><b>Competences Expected: C1,C18</b></th></tr>
<tr><th><b>Learning Outcomes</b></th><th><b>Topics</b></th></tr>
<tr><td>• Design and implement a class [Usage]
• Use subclassing to design simple class hierarchies that allow code to be reused for distinct subclasses [Usage]
• Correctly reason about control flow in a program using dynamic dispatch [Usage]
• Compare and contrast (1) the procedural/functional approach—defining a function for each operation with the function body providing a case for each data variant—and (2) the object-oriented approach—defining a class for each data variant with the class definition providing a method for each operation Understand both as defining a matrix of operations and variants [Usage]
• Explain the relationship between object-oriented inheritance (code-sharing and overriding) and subtyping (the idea of a subtype being usable in a context that expects the supertype) [Usage]
• Use object-oriented encapsulation mechanisms such as interfaces and private members [Usage]
• Define and use iterators and other operations on aggregates, including operations that take functions as arguments, in multiple programming languages, selecting the most natural idioms for each language [Usage]</td><td>• Object-oriented design
– Decomposition into objects carrying state and having behavior
– Class-hierarchy design for modeling
• Definition of classes: fields, methods, and constructors
• Subclasses, inheritance, and method overriding
• Dynamic dispatch: definition of method-call
• Subtyping
– Subtype polymorphism; implicit upcasts in typed languages
– Notion of behavioral replacement: subtypes acting like supertypes
– Relationship between subtyping and inheritance
• Object-oriented idioms for encapsulation
– Privacy and visibility of class members
– Interfaces revealing only method signatures
– Abstract base classes
• Using collection classes, iterators, and other common library components</td></tr>
<tr><td colspan="2"><b>Readings :</b> [<b>stroustrup2013</b>]</td></tr>
</table>

Unit 8: Graphs and Trees (7)	
Competences Expected: C3,C18	
Learning Outcomes	**Topics**
• Illustrate by example the basic terminology of graph theory, and some of the properties and special cases of each type of graph/tree [Usage] • Demonstrate different traversal methods for trees and graphs, including pre, post, and in-order traversal of trees [Usage] • Model a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system [Usage] • Show how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting [Usage] • Explain how to construct a spanning tree of a graph [Usage] • Determine if two graphs are isomorphic [Usage]	• Trees – Properties – Traversal strategies • Undirected graphs • Directed graphs • Weighted graphs • Spanning trees/forests • Graph isomorphism
Readings : [Nak13]	

Unit 9: Software Design (6)	
Competences Expected: CS1,C18	
Learning Outcomes	**Topics**
• Articulate design principles including separation of concerns, information hiding, coupling and cohesion, and encapsulation [Usage] • Use a design paradigm to design a simple software system, and explain how system design principles have been applied in this design [Usage] • Construct models of the design of a simple software system that are appropriate for the paradigm used to design it [Usage] • Within the context of a single design paradigm, describe one or more design patterns that could be applicable to the design of a simple software system [Usage] • For a simple system suitable for a given scenario, discuss and select an appropriate design paradigm [Usage] • Create appropriate models for the structure and behavior of software products from their requirements specifications [Usage] • Explain the relationships between the requirements for a software product and its design, using appropriate models [Usage] • For the design of a simple software system within the context of a single design paradigm, describe the software architecture of that system [Usage] • Given a high-level design, identify the software architecture by differentiating among common software architectures such as 3-tier, pipe-and-filter, and client-server [Usage] • Investigate the impact of software architectures selection on the design of a simple system [Usage] • Apply simple examples of patterns in a software design [Usage] • Describe a form of refactoring and discuss when it may be applicable [Usage] • Select suitable components for use in the design of a software product [Usage] • Explain how suitable components might need to be adapted for use in the design of a software product [Usage] • Design a contract for a typical small software component for use in a given system [Usage] • Discuss and select appropriate software architecture for a simple system suitable for a given scenario [Usage] • Apply models for internal and external qualities in designing software components to achieve an acceptable tradeoff between conflicting quality aspects [Us-	• System design principles: levels of abstraction (architectural design and detailed design), separation of concerns, information hiding, coupling and cohesion , re-use of standard structures • Design Paradigms such as structured design (top-down functional decomposition), object-oriented analysis and design, event driven design, component-level design, data-structured centered, aspect oriented, function oriented, service oriented • Structural and behavioral models of software designs • Design patterns • Relationships between requirements and designs: transformation of models, design of contracts, invariants • Software architecture concepts and standard architectures (e.g. client-server, n-layer, transform centered, pipes-and-filters) • The use of component desing: component selection, design, adaptation and assembly of components, component and patterns, components and objects (for example, building a GUI using a standar widget set) • Refactoring designs using design patterns • Internal design qualities, and models for them: efficiency and performance, redundacy and fault tolerance, traceability of requeriments • Measurement and analysis of design quality • Tradeoffs between different aspects of quality • Application frameworks • Middleware: the object-oriented paradigm within middleware, object request brokers and marshalling, transaction processing monitors, workflow systems • Principles of secure design and coding – Principle of least privilege – Principle of fail-safe defaults – Principle of psychological acceptability

Unit 10: Requirements Engineering (1)	
Competences Expected: CS1,C18	
Learning Outcomes	**Topics**
• List the key components of a use case or similar description of some behavior that is required for a system [Usage] • Describe how the requirements engineering process supports the elicitation and validation of behavioral requirements [Usage] • Interpret a given requirements model for a simple software system [Usage] • Describe the fundamental challenges of and common techniques used for requirements elicitation [Usage] • List the key components of a data model (eg, class diagrams or ER diagrams) [Usage] • Identify both functional and non-functional requirements in a given requirements specification for a software system [Usage] • Conduct a review of a set of software requirements to determine the quality of the requirements with respect to the characteristics of good requirements [Usage] • Apply key elements and common methods for elicitation and analysis to produce a set of software requirements for a medium-sized software system [Usage] • Compare the plan-driven and agile approaches to requirements specification and validation and describe the benefits and risks associated with each [Usage] • Use a common, non-formal method to model and specify the requirements for a medium-size software system [Usage] • Translate into natural language a software requirements specification (eg, a software component contract) written in a formal specification language [Usage] • Create a prototype of a software system to mitigate risk in requirements [Usage] • Differentiate between forward and backward tracing and explain their roles in the requirements validation process [Usage]	• Describing functional requirements using, for example, use cases or users stories • Properties of requirements including consistency, validity, completeness, and feasibility • Software requirements elicitation • Describing system data using, for example, class diagrams or entity-relationship diagrams • Non functional requirements and their relationship to software quality • Evaluation and use of requirements specifications • Requirements analysis modeling techniques • Acceptability of certainty / uncertainty considerations regarding software / system behavior • Prototyping • Basic concepts of formal requirements specification • Requirements specification • Requirements validation • Requirements tracing
Readings : [stroustrup2013]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS2201. Computer Architecture (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 2 HT; 2 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Den05] Peter J. Denning. "The locality principle". In: *Commun. ACM* 48.7 (July 2005), pp. 19–24. ISSN: 0001-0782. DOI: 10.1145/1070838.1070856. URL: http://doi.acm.org/10.1145/1070838.1070856.

[Don06] J. Dongarra. "Trends in high performance computing: a historical overview and examination of future developments". In: *Circuits and Devices Magazine, IEEE* 22.1 (2006), pp. 22–27. ISSN: 8755-3996. DOI: 10.1109/MCD.2006.1598076.

[EA05] Hesham El-Rewini and Mostafa Abd-El-Barr. *Advanced Computer Architecture and Parallel Processing.* Hoboken, NJ: John Wiley & Sons, 2005. ISBN: 0-471-46740-5.

[HP06] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach.* 4th. San Mateo, CA: Morgan Kaufman, 2006.

[Joh91] M. Johnson. *Superscalar microprocessor design.* Prentice Hall series in innovative technology. Prentice Hall, 1991. ISBN: 9780138756345.

[Par02] Behrooz Parhami. *Introduction to parallel processing: algorithms and architectures.* Plenum series in computer science. Plenum Press, 2002. ISBN: 9780306459702.

[Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers.* New York: Oxford Univ. Press, 2005. ISBN: ISBN 0-19-515455-X.

[PH04] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface.* 3rd ed. San Mateo, CA: Morgan Kaufman, 2004.

[Sta10] William Stalings. *Computer Organization and Architecture: Designing for Performance.* 8th. Upper Saddle River, NJ: Prentice Hall, 2010.

6. **Information about the course**

(a) **Brief description about the course** It is necessary that the professional in Computer Science has a solid knowledge of the organization and operation of the various computer systems in which the programming environment is installed. This will also know how to establish the scope and limits of the applications that are developed according to the platform being used.

The following topics will be addressed: basic digital logic components in a computer system, design of instruction sets, microarchitecture of the processor and execution in pipelining, organization of memory: cache and virtual memory, protection and sharing, system I / O and interrupts, super-scalar architectures and out-of-order execution, vector computers, multithreading architectures, symmetric multiprocessors, memory and synchronization models, integrated systems and parallel computers.

(b) **Prerrequisites:** CS1D02. Discrete Structures II. (2^{nd} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- This course is intended to provide the student with a solid foundation in the evolution of computer architectures and the factors that influenced the design of hardware and software in today's computer systems.
- Ensure understanding of what hardware is itself and how it interacts with hardware and software in a current computing system.
- To deal with the following topics: basic digital logic components in a computer system, design of instruction sets, microarchitecture of the processor and execution in pipelining, organization of memory: cache and virtual memory, protection and sharing, system I / O and interrupts, super-scalar architectures and out-of-order execution, vector computers, multithreading architectures, symmetric multiprocessors, memory and synchronization models, integrated systems and parallel computers.

8. Contribution to Outcomes

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

9. Competences (IEEE)

C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.⇒ **Outcome i**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome b,i,g**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome b,g**

C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.⇒ **Outcome i**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome b,i,g**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome b,g**

10. List of topics

1. Digital logic and digital systems
2. Machine level representation of data
3. Assembly level machine organization
4. Functional organization
5. Memory system organization and architecture
6. Interfacing and communication
7. Multiprocessing and alternative architectures
8. Performance enhancements

11. Methodology and Evaluation

Methodology:

Theory Sessions:

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:

In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:

Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Digital logic and digital systems (18)	
Competences Expected: C8	
Learning Outcomes	**Topics**
• Describe the progression of computer technology components from vacuum tubes to VLSI, from mainframe computer architectures to the organization of warehouse-scale computers [Familiarity] • Comprehend the trend of modern computer architectures towards multi-core and that parallelism is inherent in all hardware systems [Usage] • Explain the implications of the “power wall” in terms of further processor performance improvements and the drive towards harnessing parallelism [Usage] • Articulate that there are many equivalent representations of computer functionality, including logical expressions and gates, and be able to use mathematical expressions to describe the functions of simple combinational and sequential circuits [Familiarity] • Design the basic building blocks of a computer: arithmetic-logic unit (gate-level), registers (gate-level), central processing unit (register transfer-level), memory (register transfer-level) [Usage] • Use CAD tools for capture, synthesis, and simulation to evaluate simple building blocks (eg, arithmetic-logic unit, registers, movement between registers) of a simple computer design [Familiarity] • Evaluate the functional and timing diagram behavior of a simple processor implemented at the logic circuit level [Assessment]	• Overview and history of computer architecture • Combinational vs. sequential logic/Field programmable gate arrays as a fundamental combinational + sequential logic building block • Multiple representations/layers of interpretation (hardware is just another layer) • Computer-aided design tools that process hardware and architectural representations • Register transfer notation/Hardware Description Language (Verilog/VHDL) • Physical constraints (gate delays, fan-in, fan-out, energy/power)
Readings : [Par05], [PH04]	

Unit 2: Machine level representation of data (8)	
Competences Expected: C9	
Learning Outcomes	**Topics**
• Explain why everything is data, including instructions, in computers [Assessment] • Explain the reasons for using alternative formats to represent numerical data [Familiarity] • Describe how negative integers are stored in sign-magnitude and twos-complement representations [Usage] • Explain how fixed-length number representations affect accuracy and precision [Usage] • Describe the internal representation of non-numeric data, such as characters, strings, records, and arrays [Usage] • Convert numerical data from one format to another [Usage]	• Bits, bytes, and words • Numeric data representation and number bases • Fixed- and floating-point systems • Signed and twos-complement representations • Representation of non-numeric data (character codes, graphical data) • Representation of records and arrays
Readings : [Par05], [Sta10]	

Unit 3: Assembly level machine organization (8)	
Competences Expected: C4,CS3	
Learning Outcomes	**Topics**
• Explain the organization of the classical von Neumann machine and its major functional units [Familiarity] • Describe how an instruction is executed in a classical von Neumann machine, with extensions for threads, multiprocessor synchronization, and SIMD execution [Familiarity] • Describe instruction level parallelism and hazards, and how they are managed in typical processor pipelines [Familiarity] • Summarize how instructions are represented at both the machine level and in the context of a symbolic assembler [Familiarity] • Demonstrate how to map between high-level language patterns into assembly/machine language notations [Usage] • Explain different instruction formats, such as addresses per instruction and variable length vs fixed length formats [Usage] • Explain how subroutine calls are handled at the assembly level [Usage] • Explain the basic concepts of interrupts and I/O operations [Familiarity] • Write simple assembly language program segments [Usage] • Show how fundamental high-level programming constructs are implemented at the machine-language level [Usage]	• Basic organization of the von Neumann machine • Control unit; instruction fetch, decode, and execution • Instruction sets and types (data manipulation, control, I/O) • Assembly/machine language programming • Instruction formats • Addressing modes • Subroutine call and return mechanisms • I/O and interrupts • Heap vs. Static vs. Stack vs. Code segments
Readings : [Par05], [PH04], [HP06]	

Unit 4: Functional organization (8)	
Competences Expected: C9	
Learning Outcomes	**Topics**
• Compare alternative implementation of datapaths [Assessment] • Discuss the concept of control points and the generation of control signals using hardwired or microprogrammed implementations [Familiarity] • Explain basic instruction level parallelism using pipelining and the major hazards that may occur [Usage] • Design and implement a complete processor, including datapath and control [Usage] • Determine, for a given processor and memory system implementation, the average cycles per instruction [Assessment]	• Implementation of simple datapaths, including instruction pipelining, hazard detection and resolution • Control unit: hardwired realization vs. microprogrammed realization • Instruction pipelining • Introduction to instruction-level parallelism (ILP)
Readings : [Par05], [HP06]	

Unit 5: Memory system organization and architecture (8)	
Competences Expected: CS3	
Learning Outcomes	**Topics**
• Identify the main types of memory technology (eg, SRAM, DRAM, Flash, magnetic disk) and their relative cost and performance [Familiarity] • Explain the effect of memory latency on running time [Familiarity] • Describe how the use of memory hierarchy (cache, virtual memory) is used to reduce the effective memory latency [Usage] • Describe the principles of memory management [Usage] • Explain the workings of a system with virtual memory management [Usage] • Compute Average Memory Access Time under a variety of cache and memory configurations and mixes of instruction and data references [Assessment]	• Storage systems and their technology • Memory hierarchy: importance of temporal and spatial locality • Main memory organization and operations • Latency, cycle time, bandwidth, and interleaving • Cache memories (address mapping, block size, replacement and store policy) • Multiprocessor cache consistency/Using the memory system for inter-core synchronization/atomic memory operations • Virtual memory (page table, TLB) • Fault handling and reliability • Error coding, data compression, and data integrity
Readings : [Par05], [PH04], [Den05]	

Unit 6: Interfacing and communication (8)	
Competences Expected: C4,C9,CS3	
Learning Outcomes	**Topics**
• Explain how interrupts are used to implement I/O control and data transfers [Familiarity] • Identify various types of buses in a computer system [Familiarity] • Describe data access from a magnetic disk drive [Usage] • Compare common network organizations, such as ethernet/bus, ring, switched vs routed [Assessment] • Identify the cross-layer interfaces needed for multimedia access and presentation, from image fetch from remote storage, through transport over a communications network, to staging into local memory, and final presentation to a graphical display [Familiarity] • Describe the advantages and limitations of RAID architectures [Familiarity]	• I/O fundamentals: handshaking, buffering, programmed I/O, interrupt-driven I/O • Interrupt structures: vectored and prioritized, interrupt acknowledgment • External storage, physical organization, and drives • Buses: bus protocols, arbitration, direct-memory access (DMA) • Introduction to networks: communications networks as another layer of remote access • Multimedia support • RAID architectures
Readings : [Par05], [Sta10]	

Unit 7: Multiprocessing and alternative architectures (8)	
Competences Expected: C9	
Learning Outcomes	**Topics**
• Discuss the concept of parallel processing beyond the classical von Neumann model [Assessment] • Describe alternative parallel architectures such as SIMD and MIMD [Familiarity] • Explain the concept of interconnection networks and characterize different approaches [Usage] • Discuss the special concerns that multiprocessing systems present with respect to memory management and describe how these are addressed [Familiarity] • Describe the differences between memory backplane, processor memory interconnect, and remote memory via networks, their implications for access latency and impact on program performance [Assessment]	• Power Law • Example SIMD and MIMD instruction sets and architectures • Interconnection networks (hypercube, shuffle-exchange, mesh, crossbar) • Shared multiprocessor memory systems and memory consistency • Multiprocessor cache coherence
Readings : [Par05], [Par02], [EA05]	

Unit 8: Performance enhancements (8)	
Competences Expected: C8,C9	
Learning Outcomes	**Topics**
• Describe superscalar architectures and their advantages [Familiarity] • Explain the concept of branch prediction and its utility [Usage] • Characterize the costs and benefits of prefetching [Assessment] • Explain speculative execution and identify the conditions that justify it [Assessment] • Discuss the performance advantages that multithreading offered in an architecture along with the factors that make it difficult to derive maximum benefits from this approach [Assessment] • Describe the relevance of scalability to performance [Assessment]	• Superscalar architecture • Branch prediction, Speculative execution, Out-of-order execution • Prefetching • Vector processors and GPUs • Hardware support for multithreading • Scalability • Alternative architectures, such as VLIW/EPIC, and Accelerators and other kinds of Special-Purpose Processors
Readings : [Par05], [Par02], [PH04], [Don06], [Joh91]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: CS2B01. Platform Based Development (Mandatory)
2. Credits: 2
3. Hours of theory and Lab: 1 HT; 2 HL; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[ADC13] J. Annuzzi, L. Darcey, and S. Conder. *Introduction to Android Application Development: Android Essentials.* Developer's Library. Pearson Education, 2013. ISBN: 9780133477337.

[Fie00] Roy Thomas Fielding. "Fielding dissertation: Chapter 5: Representational state transfer (rest)". In: *http://www. ics. uci. edu/˜ fielding/pubs/dissertation/rest_arch_style. htm* (2000).

[FR11] Eric Freeman and Elisabeth Robson. *Head first HTML5 programming: building web apps with JavaScript.* " O'Reilly Media, Inc.", 2011.

[Gro09] R. Grove. *Web Based Application Development.* Jones & Bartlett Learning, 2009. ISBN: 9780763759407.

[Mar17] Robert C Martin. *Clean architecture: a craftsman's guide to software structure and design.* Prentice Hall Press, 2017.

6. Information about the course

(a) **Brief description about the course** The world has changed due to the use of fabric and related technologies, rapid, timely and personalized access to the information, through web technology, ubiquitous and pervasive; they have changed the way we do things, how do we think? and how does the industry develop? Web technologies, ubiquitous and pervasive are based on the development of web services, web applications and mobile applications, which are necessary to understand the architecture, design, and implementation of web services, web applications and mobile applications.

(b) **Prerrequisites:** CS1102. Objects oriented programming I. (2^{nd} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- That the student is able to design and implement services, web applications using tools and languages such as HTML, CSS, JavaScript (including AJAX), back-end scripting and a database, at an intermediate level.
- That the student is able to develop mobile applications, administration of web servers in a Unix system and an introduction to web security, at an intermediate level.

8. Contribution to Outcomes

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome c,d,i**

C6. Ability to design and implement larger structural units that utilize algorithms and data structures and the interfaces through which these units communicate.⇒ **Outcome c,d,i**

CS8. Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, multimedia systems and mobile systems..⇒ **Outcome g**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome c,d,i**

C6. Ability to design and implement larger structural units that utilize algorithms and data structures and the interfaces through which these units communicate.⇒ **Outcome c,d,i**

CS8. Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, multimedia systems and mobile systems..⇒ **Outcome g**

10. List of topics

1. Introduction
2. Web Platforms
3. Desarrollo de servicios y aplicaciones web
4. Mobile Platforms
5. Mobile Applications for Android Handheld Systems

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Introduction (5)	
Competences Expected: CS8	
Learning Outcomes	**Topics**
• Describe how platform-based development differs from general purpose programming [Familiarity] • List characteristics of platform languages [Familiarity] • Write and execute a simple platform-based program [Familiarity] • List the advantages and disadvantages of programming with platform constraints [Familiarity]	• Overview of platforms (e.g., Web, Mobile, Game, Industrial) • Programming via platform-specific APIs • Overview of Platform Languages (e.g., Objective C, HTML5) • Programming under platform constraints
Readings : [Fie00], [Gro09], [ADC13]	

Unit 2: Web Platforms (5)	
Competences Expected: C1,C6	
Learning Outcomes	**Topics**
• Design and Implement a simple web application [Familiarity] • Describe the constraints that the web puts on developers [Familiarity] • Compare and contrast web programming with general purpose programming [Familiarity] • Describe the differences between Software-as-a-Service and traditional software products [Familiarity] • Discuss how web standards impact software development [Familiarity] • Review an existing web application against a current web standard [Familiarity]	• Web programming languages (e.g., HTML5, Java Script, PHP, CSS) • • Web Platform constraints: Client-Server, Stateless-Stateful, Cache, Uniform Interface, Layered System, Code on Demand, ReST. • Web platform constraints • Software as a Service (SaaS) • Web standards
Readings : [Fie00]	

Unit 3: Desarrollo de servicios y aplicaciones web (25)	
Competences Expected: C1,C6	
Learning Outcomes	**Topics**
• Server-side python scripting language: variables, data types, operations, strings, functions, control statements, arrays, files and directory access, maintain state. [Usage] • Web programming approach using embedded python. [Usage] • Accessing and Manipulating MySQL. [Usage] • The Ajax web application development approach. [Usage] • DOM and CSS used in JavaScript. [Usage] • Asynchronous Content Update Technologies. [Usage] • XMLHttpRequest objects use to communicate between clients and servers. [Usage] • XML and JSON. [Usage] • XSLT and XPath as mechanisms for transforming XML documents. [Usage] • Web services and APIs (especially Google Maps). [Usage] • Macros Ajax for the development of contemporary web applications. [Usage] • Design patterns used in web applications. [Usage]	• Describe, identify and debug issues related to web application development • Design and development of interactive web applications using HTML5 and Python • Use MySQL for data management and manipulate MySQL with Python • Design and development of asynchronous web applications using Ajax techniques • Using dynamic client side Javascript scripting language and server side python scripting language with Ajax • Apply XML / JSON technologies for data management with Ajax • Use framework, services and Ajax web APIs and apply design patterns to web application development
Readings : [FR11]	

Unit 4: Mobile Platforms (5)	
Competences Expected: C1,C6	
Learning Outcomes	**Topics**
• Design and implement a mobile application for a given mobile platform [Familiarity] • Discuss the constraints that mobile platforms put on developers [Familiarity] • Discuss the performance vs power tradeoff [Familiarity] • Compare and Contrast mobile programming with general purpose programming [Familiarity]	• Mobile programming languages • Design Principles: Segregation of Interfaces, Single Responsability, Separation of concerns, Dependency Inversion. • Challenges with mobility and wireless communication • Location-aware applications • Performance / power tradeoffs • Mobile platform constraints • Emerging technologies
Readings : [Mar17], [ADC13]	

Unit 5: Mobile Applications for Android Handheld Systems (25)	
Competences Expected: C1,C6	
Learning Outcomes	**Topics**
• Students identify necessary software and install it on their personal computers. • Students perform various tasks to familiarize themselves with the Android platform and Environment for development. [Usage] • Students build applications that trace the lifecycle callback methods emitted by the Android platform and demonstrate the behavior of Android when device configuration changes (for example, when the device moves from vertical to horizontal and vice versa). [Usage] • Students build applications that require starting multiple activities through both standard and custom methods. [Usage] • Students build applications that require standard and custom permissions. [Usage] • Students build an application that uses a single code base, but creates different user interfaces depending on the screen size of a device. [Usage] • Students construct a to-do list manager using the user interface elements discussed in class. The application allows users to create new items and to display them in a ListView. [Usage] • Students build an application that uses location information to collect latitude, length of places they visit. [Usage]	• The Android Platform • The Android Development Environment • Application Fundamentals • The Activity Class • The Intent Class • Permissions • The Fragment Class • User Interface Classes • User Notifications • The BroadcastReceiver Class • Threads, AsyncTask & Handlers • Alarms • Networking (http class) • Multi-touch & Gestures • Sensors • Location & Maps
Readings : [ADC13]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: EN0021. Physics II (Mandatory)
2. Credits: 4
3. Hours of theory and Lab: 4 HT; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[EL98] Robert Eisberg and Lawrence Lerner. *Física: Fundamentos y Aplicaciones*. Vol. 1. Mc Graw Hill, 1998.

[Gia84] Douglas C. Giancoli. *General Physics*. Prentice Hall, Inc., 1984.

[Ray98] Serway Raymond. *Física*. Vol. 1. Mc Graw Hill, 1998.

[RH98] Robert Resnick and David Halliday. *Física para Estudiantes de Ciencias e Ingeniería*. John Wiley, 1998.

[Sea98] Francis Sears. *Física Universitaria*. Addison Wesley-Longman, 1998.

[Tip98] Paul Tipler. *Física*. 3rd ed. Editorial Reverte, 1998.

6. Information about the course

(a) **Brief description about the course** Show a high degree of mastery of the laws of wave motion, the nature of fluids, and thermodynamics. Using properly the concepts of wave movement, fluids and thermodynamics in solving problems of daily life. Possess ability and ability in the interpretation of wave, fluid and thermodynamic phenomena, which contribute to the development of efficient and useful solutions in different areas of computer science.

(b) **Prerrequisites:** ME0019. Physics I. (2^{nd} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- That the student learn and master fluent principles static and moving.
- That the student learn and master the principles of MAS, particularly the wave movement.
- Thar the student learn and master the principles of thermodynamics.
- That the student learn to apply principles of the Physics of fluids, waves and thermodynamics to develop computational models

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome i,j**

10. List of topics

1. FI1. Elasticidad
2. FI2. Fluidos
3. FI3. Movimiento Periódico
4. FI4. Ondas
5. FI5. Temperatura y Teoría Cinética
6. FI6. Calor y primera Ley de la Termodinámica
7. FI7. Máquinas térmicas, entropía y la segunda ley de la Termodinámica

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: FI1. Elasticidad (4)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Understand and characterize the processes of elasticity • Solve problems	• Effort and unit deformation • Young's Module • Poisson Module and Coefficient • Stiffness Module • Module and coefficient of compressibility
Readings : [Sea98], [EL98]	

Unit 2: FI2. Fluidos (8)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Explain, analyze and characterize fluid pressure • Understand, characterize and apply the principle of Archimedes • Understand, characterize and apply the Bernoulli principle • Explain, analyze and characterize surface tension and capillarity	• Density and specific gravity • Pressure in fluids. Atmospheric pressure and gauge pressure • Principle of Pascal. Pressure measurement: manometer and barometer • Buoyancy and Principle of Archimedes • Flowing Fluids: Flow and Continuity Equation • Bernoulli equation. Applications of the Bernoulli principle: Torricelli's theorem, the ventura tube • Surface tension and capillarity
Readings : [Ray98], [Tip98]	

Unit 3: FI3. Movimiento Periódico (8)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Explain, analyze and characterize the oscillatory movement from the MAS. • Solve problems.	• Introduction .Elastic modulus of a Spring • Simple harmonic motion. Energy in simple harmonic oscillator • Reference circle: the period and the sinusoidal nature of simple harmonic motion • Simple pendulum. • Cushioned harmonic motion. • Forced oscillations: resonance.
Readings : [Sea98], [Ray98]	

Unit 4: FI4. Ondas (8)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Explain, find and characterized through problems of the daily life the undulatory movement, as well as, the reflection and transmission of waves in the space. • Solve problems	• Wave motion. Types of waves. One-dimensional traveling waves • Wave Overlay and Interference • Velocity of the waves in a tight rope. Reflection and transmission of waves • Sine waves. Energy transmitted by sinusoidal waves in strings • Stationary waves on a rope. Sound waves. Speed of sound waves • Periodic sound waves. Intensity of periodic sound waves • Sources of sound: vibratory strings and vibrating air columns • Doppler Effect
Readings : [EL98], [RH98], [Gia84]	

Unit 5: FI5. Temperatura y Teoría Cinética (12)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Explain, analyze and characterize the concept of Temperature and the thermal expansion of solids and liquids • Understanding the ideal gas law and the isothermal and adiabatic processes for an ideal gas • Understand the zero law of thermodynamics • Solve problems	• Atoms. Temperature. Thermometers and temperature scales • Thermal expansion of solids and liquids. Coefficients of linear, surface and cubic expansion • Laws of gases and absolute temperature. The ideal gas law in molecular terms: Avogadro's number • Kinetic theory and molecular interpretation of temperature. Distribution of molecular velocities • Isothermal and adiabatic processes for an ideal gas. The equipartition of energy • Termodinámica. Tipos de sistemas que estudia la Termodinámica • Zero Law of Thermodynamics • The constant-volume gas thermometer and the Kelvin scale • Punto triple del agua
Readings : [EL98], [RH98]	

Unit 6: FI6. Calor y primera Ley de la Termodinámica (8)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Understand the concept of heat and internal energy of an ideal gas • Explain, analyze and characterize the first law of thermodynamics • Solve problems	• Heat as energy transfer • Heat capacity and specific heat • Internal energy of an ideal gas • Specific heat of an ideal gas • Phase changes. Latent heat of fusion and vaporization • Calorimetry. Work and heat in thermodynamic processes • The first law of thermodynamics • Some applications of the first law of thermodynamics • Transmission of heat by conduction, convection and radiation
Readings : [EL98], [RH98]	

Unit 7: FI7. Máquinas térmicas, entropía y la segunda ley de la Termodinámica (8)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Explain, analyze and characterize the first law of thermodynamics • Explain, analyze and characterize the Carnot machine • Solve problems	• Thermal Machines and the Second Law of Thermodynamics • Reversible and irreversible processes. The Carnot Machine • Absolute temperature range.Chillers • Entropy. Entropy changes in irreversible processes
Readings : [EL98], [RH98]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: GH0008. Business Management (Mandatory)
2. Credits: 2
3. Hours of theory and Lab: 1 HT; 2 HP; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[A12] Maurya. A. *Running lean: Iterate from plan A to a plan that works.* Sebastopol, 2012.

[PF03] Kotler. P and Trias de Bes. F. *Marketing Lateral.* Madrid,Person Prentice Hill., 2003.

6. Information about the course

(a) **Brief description about the course** .

(b) **Prerrequisites:** GH0007. Introduction to Business Development. (2^{nd} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Understanding the basics of the business planning process and its role within the business life cycle.

8. Contribution to Outcomes

f) An ability to communicate effectively. (**Usage**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

n) Apply knowledge of the humanities in their professional work. (**Usage**)

9. Competences (IEEE)

C17. Ability to properly express in oral and written media as expected from a university graduate. ⇒ **Outcome f,h,n**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome f,n**

C24. Understanding the need for lifelong learning and improving skills and abilities.⇒ **Outcome f,h**

10. List of topics

1. Business Management

11. Methodology and Evaluation
Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

Unit 1: Business Management (16)	
Competences Expected: C17,C20	
Learning Outcomes	**Topics**
• Understood the importance of effective planning and how it contributes to a company has launch and success.	• The business lifecycle: why do I need a business plan? • Differences between business model and business planning. • The importance of a well-structured team. • Environmental analysis and main market research tools. • Strategic Planning: Why it is necessary and how it is done. • The importance of capital: human, financial, and intellectual. • How to build an operations plan • The basics of marketing: defining marketing strategy. • Financial Projections : costs and sales. • Legal issues. • Responsible Businesses: the basics.
Readings : [A12], [PF03]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: GH1102. English II (Mandatory)
2. Credits: 3
3. Hours of theory and Lab: 10 HP; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[Cam06] Cambridge. *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford, 2006.

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

[SJ02] Liz Soars and John. *American Headway N 1 Student Book*. Editorial Oxford, 2002.

6. Information about the course

(a) **Brief description about the course** A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of people's lives. In the case of foreign languages, English is undoubtedly the most practical because it is spoken around the world. There is no country where it is not spoken. In careers related to tourist services english is perhaps the most important practical tool that the student must master from the outset as part of his comprehensive education.

(b) **Prerrequisites:** GH1101. English I. (2nd Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Develop the ability to fluently speak the language..
- Increase vocabulary and handle simple expressions

8. Contribution to Outcomes

f) An ability to communicate effectively. (**Usage**)

9. Competences (IEEE)

C25. Ability to communicate in a second language.⇒ **Outcome f**

10. List of topics

1. How long ago?
2. Food you like!
3. The world of work
4. Looking good!
5. Life is an adventure!

6. You‘re pretty smart!
7. Have you ever?

11. Methodology and Evaluation
Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

Unit 1: How long ago? (0)	
Competences Expected: 2	
Learning Outcomes	**Topics**
• At the end of the eighth unit, each student, understanding the grammar of the past tense is able to express a greater number of expressions of time and also to use prepositions to describe varied places and times. It is also capable of analyzing and expressing ideas about dates and numbers in order.	• Simple past • Negative sentences with ago. • Conjunctions • Expressions of Time in past • Phonetic relations and symbols • Expressions to give the date
Readings : [SJ02], [Cam06], [Mac99]	

Unit 2: Food you like! (0)	
Competences Expected: 2	
Learning Outcomes	**Topics**
• At the end of the ninth unit, students having identified how to express orders and make collections, uses in various situations. Express situations and states related to quantities. Explain and apply food and drinks vocabulary.	• Accounting and Non-Accounting Nouns • Expressions with Would like and I’d like • Quantifiers • Meals around the world • Formal requests • Formal letters
Readings : [SJ02], [Cam06], [Mac99]	

<table>
<tr><th colspan="2">Unit 3: The world of work (0)</th></tr>
<tr><th colspan="2">Competences Expected: 2</th></tr>
<tr><th>Learning Outcomes</th><th>Topics</th></tr>
<tr><td>• At the end of the tenth unit, students who have recognized the characteristics of adjectives use these to make comparisons of various types. Describes people and places and directions. They will use conjunctions to unite type ideas.</td><td>• Adjectives
• Sentences with Comparative Adjectives
• Sentences with Superlative Adjectives
• Cities and countryside
• Directional indications</td></tr>
<tr><td colspan="2"><b>Readings :</b> [SJ02], [Cam06], [Mac99]</td></tr>
</table>

<table>
<tr><th colspan="2">Unit 4: Looking good! (0)</th></tr>
<tr><th colspan="2">Competences Expected: 2</th></tr>
<tr><th>Learning Outcomes</th><th>Topics</th></tr>
<tr><td>• At the end of the eleventh unit, students having identified the idea of expressing ideas of actions that occur at the time or that relate at any time structure sentences in Present Progressive. They express ideas of possession with regard to clothes and colors.</td><td>• Present continuous
• Affirmative sentences, Negatives and Questions
• Use of Whose
• Possessive pronouns
• Clothing and colors
• Expressions to wear in clothing stores
• Phonetic symbols.</td></tr>
<tr><td colspan="2"><b>Readings :</b> [SJ02], [Cam06], [Mac99]</td></tr>
</table>

<table>
<tr><th colspan="2">Unit 5: Life is an adventure! (0)</th></tr>
<tr><th colspan="2">Competences Expected: 2</th></tr>
<tr><th>Learning Outcomes</th><th>Topics</th></tr>
<tr><td>• At the end of the twelfth unit, students, from the understanding of future time, will elaborate sentences using the necessary elements. They will also assimilate the need to express purposeful infinitives. They will acquire vocabulary to describe the climate. Expressions will be presented to make and ask for suggestions.</td><td>• Use of going to
• Future time sentences
• Expressions of Quantity.
• Action verbs
• Vocabulary of the climate
• Expressions of Suggestion
• Write a postcard</td></tr>
<tr><td colspan="2"><b>Readings :</b> [SJ02], [Cam06], [Mac99]</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Unit 6: You‘re pretty smart! (0)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: 2</b></td></tr>
<tr><td><b>Learning Outcomes</b></td><td><b>Topics</b></td></tr>
<tr><td>• At the end of the thirteenth unit, students having learned the fundamentals of structuring various questions, will perform application work in appropriate contexts. They emphasize the difference between adjectives and adverbs. They describe feelings. They use expressions to catch a train. They assume the idea is suffixes and prefixes.</td><td>• Forms of Questions
• Adverbs and Adjectives
• Vocabulary description of feelings
• Expressions for train travel
• Writing Short Stories
• Readings</td></tr>
<tr><td colspan="2"><b>Readings :</b> [SJ02], [Cam06], [Mac99]</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Unit 7: Have you ever? (0)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: 2</b></td></tr>
<tr><td><b>Learning Outcomes</b></td><td><b>Topics</b></td></tr>
<tr><td>• At the end of the fourteenth unit, students having known the fundamentals of the structure of the Present Perfect experience the need to express this type of time in actions. They will practice in appropriate contexts. They emphasize the difference between simple past and perfect present. Describe actions with never, ever, and yet. They use expressions to use at an airport.</td><td>• Perfect present
• Keywords with never, ever, and yet
• Vocabulary verbs in Past participle
• Expressions for airplane travel
• Writing thank-you letters
• Readings</td></tr>
<tr><td colspan="2"><b>Readings :</b> [SJ02], [Cam06], [Mac99]</td></tr>
</table>

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** EG0006. Math III (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 4 HT; 1 HP; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[AR14] H. Anton and C. Rorres. *Elementary Linear Algebra, Applications Version.* 11th. Wiley, 2014.

[CC15] S.C. Chapra and R.P. Canale. *Numerical Methods for Engineers,* 7th. Vol. 1. McGraw-Hill, 12015.

6. Information about the course

(a) **Brief description about the course** This course introduces the first concepts of linear algebra as well as numerical methods with an emphasis on problem solving with the Scilab open source libe package. Mathematical theory is limited to fundamentals, while effective application for problem solving is privileged. In each subject, a few methods of relevance for engineering are taught. Knowledge of these methods prepares students for the search for more advanced alternatives, if required.

(b) **Prerrequisites:** EG0005. Math II. (2^{nd} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Ability to apply knowledge about Mathematics.
- Ability to apply engineering knowledge.
- Ability to apply the modern knowledge, techniques, skills and tools of modern engineering to the practice of engineering

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome j**

C24. Understanding the need for lifelong learning and improving skills and abilities.⇒ **Outcome j**

10. List of topics

1. Introduction
2. Linear Algebra
3. Numerical methods

11. Methodology and Evaluation
Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

Unit 1: Introduction (18)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Be able to understand the basic concepts and importance of Linear Algebra and Numerical Methods.	• Importance of linear algebra and numerical methods. Examples.
Readings : [AR14], [CC15]	

Unit 2: Linear Algebra (14)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Understanding the basics concepts of Linear Algebra. • Solve properly linear transformations problems.	• Elementary matrix algebra and determinants • Null space and exact solutions of systems of linear equations Ax=b: – Tridiagonal and triangular systems and Gaussian elimination with and without pivoting. – LU factorization and Crout algorithm. • Basics on eigenvalues and eigenvectors: – Characteristic polynomials. – Algebraic and geometric multiplicities. • Least squares estimation. • Linear transformations.
Readings : [AR14], [CC15]	

Unit 3: Numerical methods (22)

Competences Expected: C24

Learning Outcomes	**Topics**
• Understanding the basics concepts of Numerical Methods. • Applying the most frequent methods for the resolution of mathematical problems. • Implementing and applying numerical algorithms for the solution of mathematical problems using the Scilab open-source computational package. • Applying Scilab for the solution of mathematical problems and for plotting graphs.	• Basics on solutions of systems of linear equations Ax=b: Jacobi and Gauss Seidel methods. • Application of matrix factorizations to the solution of linear systems (singular value decomposition, QR, Cholesky) Numerical computation of null space, rank and condition number. • Root finding: – Bisection. – Fixed-point iteration. – Newton-Raphson methods. • Basics on interpolation: – Newton and Lagrange polynomial interpolations – Spline interpolation • Basics on numerical differentiation and Taylor approximation • Basics on numerical integration: – Trapezium, midpoint and Simpson rule – Gaussian quadrature • Basics on numerical solutions to ODEs: – Finite differences; Euler and Runge-Kutta methods – Converting higher order ODEs into a system of low order ODEs – Runge-Kutta methods for systems of equations – Single shooting method • Short introduction to optimization techniques: overview on linear programming, bounded linear systems, quadratic programming, gradient descent.

Readings : [AR14], [CC15]

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS2100. Algorithms and Data Structures (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Cor+09] Thomas H. Cormen et al. *Introduction to Algorithms.* Third Edition. ISBN: 978-0-262-53305-8. MIT Press, 2009.

[Fag+14] José Fager et al. *Estructura de datos.* First Edition. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIN), 2014.

6. **Information about the course**

(a) **Brief description about the course** The theoretical foundation of all branches of computing rests on algorithms and data structures, this course will provide participants with an introduction to these topics, thus forming a basis that will serve for the following courses in the career.

(b) **Prerrequisites:** CS1103. Objects oriented programming II. (3^{rd} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Make the student understand the importance of algorithms for solving problems.
- Introduce the student to the field of application of data structures.

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

9. **Competences (IEEE)**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome b**

C5. Ability to implement algorithms and data structures in software.⇒ **Outcome c**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome b**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome b**

C5. Ability to implement algorithms and data structures in software.⇒ **Outcome c**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome b**

10. List of topics

1. Grafos
2. Matrices Esparzas
3. Arboles Equilibrados

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Grafos (12)	
Competences Expected: C1,C2,C5	
Learning Outcomes	**Topics**
• Acquire Dexterity to Perform Correct Implementation. [Usage] • Develop knowledge to decide when it is better to use one implementation technique than another. [Usage]	• Graph Concept • Directed Graphs and Non-directed Graphs. • Using Graphs. • Measurement of efficiency ,in time and space. • Adjacency matrices. • Tag adjacent matrices. • Adjacency Lists. • Implementation of graphs using adjacency matrices. • Graph Implementation using adjacency lists • Insertion, search and deletion of nodes and edges. • Graph search algorithms.
Readings : [Cor+09], [Fag+14]	

Unit 2: Matrices Esparzas (8)	
Competences Expected: C1,C2,C5	
Learning Outcomes	**Topics**
• Understand the use and implementation of scatter matrices.[Assessment]	• Initial concepts. • Dense Matrices • Measurement of Efficiency in Time and Space • Static scatter vs. dynamic matrix creation. • Insert, search, and delete methods.
Readings : [Cor+09], [Fag+14]	

Unit 3: Arboles Equilibrados (16)	
Competences Expected: C2,C5,C6	
Learning Outcomes	**Topics**
• Understand the basic functions of these complex structures in order to acquire the capacity for their implementation. [Assessment]	• AVL Trees. • Measurement of Efficiency. • Simple and Composite Rotations • Insertion, deletion and search. • Trees B , B+ B* y Patricia.
Readings : [Cor+09], [Fag+14]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS2101. Theory of Computation (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

   Meetings after coordination with the professor

5. **Bibliography**

[Bro93] J. Glenn Brookshear. *Teoría de la Computación.* Addison Wesley Iberoamericana, 1993.

[HU93] John E. Hopcroft and Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación.* CECSA, 1993.

[Kel95] Dean Kelley. *Teoría de Autómatas y Lenguajes Formales.* Prentice Hall, 1995.

[Kol97] Ross Kolman Busby. *Estructuras de Matemáticas Discretas para la Computación.* Prentice Hall, 1997.

6. **Information about the course**

   (a) **Brief description about the course** This course emphasizes formal languages, computer models and computability, as well as the fundamentals of computational complexity and complete NP problems.

   (b) **Prerrequisites:** CS1D02. Discrete Structures II. (2^{nd} Sem)

   (c) **Type of Course:** Mandatory

   (d) **Modality:** Face to face

7. **Specific goals of the Course**

   - That the student learn the fundamental concepts of the theory of formal languages.

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

9. **Competences (IEEE)**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome a**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome b,j**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome a**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome b,j**

10. List of topics

1. Basic Automata Computability and Complexity
2. Advanced Computational Complexity
3. Advanced Automata Theory and Computability

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Basic Automata Computability and Complexity (20)	
Competences Expected: C9	
Learning Outcomes	**Topics**
• Discuss the concept of finite state machines [Assessment] • Design a deterministic finite state machine to accept a specified language [Assessment] • Generate a regular expression to represent a specified language [Assessment] • Explain why the halting problem has no algorithmic solution [Assessment] • Design a context-free grammar to represent a specified language [Assessment] • Define the classes P and NP [Assessment] • Explain the significance of NP-completeness [Assessment] • Explain the Church-Turing thesis and its significance [Familiarity] • Explain Rice's Theorem and its significance [Familiarity] • Provide examples of uncomputable functions [Familiarity] • Prove that a problem is uncomputable by reducing a classic known uncomputable problem to it [Familiarity]	• Finite-state machines • Regular expressions • The halting problem • Context-free grammars • Introduction to the P and NP classes and the P vs. NP problem • Introduction to the NP-complete class and exemplary NP-complete problems (e.g., SAT, Knapsack) • Turing machines, or an equivalent formal model of universal computation • Nondeterministic Turing machines • Chomsky hierarchy • The Church-Turing thesis • Computability • Rice's Theorem • Examples of uncomputable functions • Implications of uncomputability
Readings : [Kol97], [Kel95]	

Unit 2: Advanced Computational Complexity (20)	
Competences Expected: C8,C9	
Learning Outcomes	**Topics**
• Define the classes P and NP (Also appears in AL/Basic Automata, Computability, and Complexity) [Assessment] • Define the P-space class and its relation to the EXP class [Assessment] • Explain the significance of NP-completeness (Also appears in AL/Basic Automata, Computability, and Complexity) [Assessment] • Provide examples of classic NP-complete problems [Assessment] • Prove that a problem is NP-complete by reducing a classic known NP-complete problem to it [Assessment]	• Review of the classes P and NP; introduce P-space and EXP • Polynomial hierarchy • NP-completeness (Cook's theorem) • Classic NP-complete problems • Reduction Techniques
Readings : [Kel95], [HU93]	

Unit 3: Advanced Automata Theory and Computability (20)	
Competences Expected: C8	
Learning Outcomes	**Topics**
• Determine a language's place in the Chomsky hierarchy (regular, context-free, recursively enumerable) [Assessment] • Convert among equivalently powerful notations for a language, including among DFAs, NFAs, and regular expressions, and between PDAs and CFGs [Assessment]	• Sets and languages – Regular languages – Review of deterministic finite automata (DFAs) – Nondeterministic finite automata (NFAs) – Equivalence of DFAs and NFAs – Review of regular expressions; their equivalence to finite automata – Closure properties – Proving languages non-regular, via the pumping lemma or alternative means • Context-free languages – Push-down automata (PDAs) – Relationship of PDAs and context-free grammars – Properties of context-free languages
Readings : [HU93], [Bro93]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: CS2701. Databases I (Mandatory)
2. Credits: 4
3. Hours of theory and Lab: 2 HT; 4 HL; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

[Dat05] C.J. Date. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Elsevier, 2005.

[Die01] Suzanne W Dietrich. *Understanding Relational Database Query Languages, First Edition*. Prentice Hall, 2001.

[EN04] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems, Fourth Edition*. Addison Wesley, 2004.

[KS02] Henry F. Korth and Abraham Silberschatz. *Fundamentos de Base de Datos*. McGraw-Hill, 2002.

[RC04] Peter Rob and Carlos Coronel. *Database Systems: Design, Implementation and Management, Sixth Edition*. Morgan Kaufmann, 2004.

[SW04] Graeme Simsion and Graham Witt. *Data Modeling Essentials, Third Edition*. Morgan Kaufmann, 2004.

[WM01] Mark Whitehorn and Bill Marklyn. *Inside Relational Databases, Second Edition*. Springer, 2001.

6. Information about the course

(a) **Brief description about the course** Information management (IM) plays a major role in almost all areas where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of accessing and updating stored information, data modeling and abstraction, and physical file storage techniques.It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which (IM) methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable restrictions, including Scalability and usability.

(b) **Prerrequisites:**

- CS1102. Objects oriented programming I. (2^{nd} Sem)
- CS1D02. Discrete Structures II. (2^{nd} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- That the student learn to represent information in a database prioritizing the efficiency in the recovery of the same.
- That the student learn the fundamental concepts of the management of databases. This includes the design of databases, database languages and the realization of databases.
- Discuss the database model with the base in relational algebra, relational calculus and the study of SQL statements.

8. Contribution to Outcomes

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome b**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome d**

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome i**

CS4. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.⇒ **Outcome i**

CS5. Specify, design, and implement computer-based systems.⇒ **Outcome j**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome b**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome d**

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome i**

CS4. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.⇒ **Outcome i**

CS5. Specify, design, and implement computer-based systems.⇒ **Outcome j**

10. List of topics

1. Database Systems
2. Data Modeling
3. Indexing
4. Relational Databases
5. Query Languages

11. Methodology and Evaluation

Methodology:

Theory Sessions:

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:

In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:

Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Database Systems (14)	
Competences Expected: C1,C7,CS4,CS5	
Learning Outcomes	**Topics**
• Explain the characteristics that distinguish the database approach from the approach of programming with data files [Usage] • Describe the most common designs for core database system components including the query optimizer, query executor, storage manager, access methods, and transaction processor [Usage] • Cite the basic goals, functions, and models of database systems [Usage] • Describe the components of a database system and give examples of their use [Usage] • Identify major DBMS functions and describe their role in a database system [Usage] • Explain the concept of data independence and its importance in a database system [Usage] • Use a declarative query language to elicit information from a database [Usage] • Describe facilities that datatbases provide supporting structures and/or stream (sequence) data, eg, text [Usage] • Describe major approaches to storing and processing large volumes of data [Usage]	• Approaches to and evolution of database systems • Components of database systems • Design of core DBMS functions (e.g., query mechanisms, transaction management, buffer management, access methods) • Database architecture and data independence • Use of a declarative query language • Systems supporting structured and/or stream content • Approaches for managing large volumes of data (e.g., noSQL database systems, use of MapReduce).
Readings : [RC04], [EN04], [Dat05], [KS02]	

Unit 2: Data Modeling (14)	
Competences Expected: C1,C2,C7,CS4,CS5	
Learning Outcomes	**Topics**
• Compare and contrast appropriate data models, including internal structures, for different types of data [Usage] • Describe concepts in modeling notation (eg, Entity-Relation Diagrams or UML) and how they would be used [Usage] • Define the fundamental terminology used in the relational data model [Usage] • Describe the basic principles of the relational data model [Usage] • Apply the modeling concepts and notation of the relational data model [Usage] • Describe the main concepts of the OO model such as object identity, type constructors, encapsulation, inheritance, polymorphism, and versioning [Usage] • Describe the differences between relational and semi-structured data models [Usage] • Give a semi-structured equivalent (eg, in DTD or XML Schema) for a given relational schema [Usage]	• Data modeling • Conceptual models (e.g., entity-relationship, UML diagrams) • Spreadsheet models • Relational data models • Object-oriented models • Semi-structured data model (expressed using DTD or XML Schema, for example)
Readings : [SW04], [EN04], [KS02]	

Unit 3: Indexing (4)	
Competences Expected: CS4,CS5	
Learning Outcomes	**Topics**
• Generate an index file for a collection of resources [Usage] • Explain the role of an inverted index in locating a document in a collection [Usage] • Explain how stemming and stop words affect indexing [Usage] • Identify appropriate indices for given relational schema and query set [Usage] • Estimate time to retrieve information, when indices are used compared to when they are not used [Usage] • Describe key challenges in web crawling, eg, detecting duplicate documents, determining the crawling frontier [Usage]	• The impact of indices on query performance • The basic structure of an index • Keeping a buffer of data in memory • Creating indexes with SQL • Indexing text • Indexing the web (e.g., web crawling)
Readings : [WM01], [Dat05], [KS02]	

Unit 4: Relational Databases (14)	
Competences Expected: 5	
Learning Outcomes	**Topics**
• Prepare a relational schema from a conceptual model developed using the entity- relationship model [Usage] • Explain and demonstrate the concepts of entity integrity constraint and referential integrity constraint (including definition of the concept of a foreign key) [Usage] • Demonstrate use of the relational algebra operations from mathematical set theory (union, intersection, difference, and Cartesian product) and the relational algebra operations developed specifically for relational databases (select (restrict), project, join, and division) [Usage] • Write queries in the relational algebra [Usage] • Write queries in the tuple relational calculus [Usage] • Determine the functional dependency between two or more attributes that are a subset of a relation [Usage] • Connect constraints expressed as primary key and foreign key, with functional dependencies [Usage] • Compute the closure of a set of attributes under given functional dependencies [Usage] • Determine whether a set of attributes form a superkey and/or candidate key for a relation with given functional dependencies [Usage] • Evaluate a proposed decomposition, to say whether it has lossless-join and dependency-preservation [Usage] • Describe the properties of BCNF, PJNF, 5NF [Usage] • Explain the impact of normalization on the efficiency of database operations especially query optimization [Usage] • Describe what is a multi-valued dependency and what type of constraints it specifies [Usage]	• Mapping conceptual schema to a relational schema • Entity and referential integrity • Relational algebra and relational calculus • Relational Database design • Functional dependency • Decomposition of a schema; lossless-join and dependency-preservation properties of a decomposition • Candidate keys, superkeys, and closure of a set of attributes • Normal forms (BCNF) • Multi-valued dependency (4NF) • Join dependency (PJNF, 5NF) • Representation theory
Readings : [WM01], [Dat05], [KS02]	

Unit 5: Query Languages (12)	
Competences Expected: C1,CS4,CS5	
Learning Outcomes	**Topics**
• Create a relational database schema in SQL that incorporates key, entity integrity, and referential integrity constraints [Usage] • Use SQL to create tables and retrieve (SELECT) information from a database [Usage] • Evaluate a set of query processing strategies and select the optimal strategy [Usage] • Create a non-procedural query by filling in templates of relations to construct an example of the desired query result [Usage] • Embed object-oriented queries into a stand-alone language such as C++ or Java (eg, SELECT Col-Method() FROM Object) [Usage] • Write a stored procedure that deals with parameters and has some control flow, to provide a given functionality [Usage]	• Overview of database languages • SQL (data definition, query formulation, update sublanguage, constraints, integrity) • Selections • Projections • Select-project-join • Aggregates and group-by • Subqueries • QBE and 4th-generation environments • Different ways to invoke non-procedural queries in conventional languages • Introduction to other major query languages (e.g., XPATH, SPARQL) • Stored procedures
Readings : [Die01], [EN04], [Cel05], [KS02]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: IN0054. Statistics and Probabilities (Mandatory)
2. Credits: 4
3. Hours of theory and Lab: 3 HT; 2 HL; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[Men14] Beaver Mendenhall. *Introducción a la probabilidad y estadística*. 13th. Cengage Learning, 2014.

[MRo14] Sheldon M.Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. 5th. Academic Press, 2014.

6. Information about the course

(a) **Brief description about the course** It provides an introduction to probability theory and statistical inference with applications, needs in data analysis, design of random models and decision making.

(b) **Prerrequisites:** EG0003. Mathematics I. (1^{st} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- An ability to design and conduct experiments, as well as to analyze and interpret data.
- An ability to identify, formulate, and solve real problems.

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome j**

10. List of topics

1. Variable Type
2. Descriptive Statistics
3. Inferential Statistics

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Variable Type (6)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Classify the relevant variables identified according to their type: continuous (interval and ratio), categorical (nominal, ordinal, dichotomous). • Identify the relevant variables of a system using a process approach.	• Variable Type: Continuous, discrete
Readings : [MRo14], [Men14]	

Unit 2: Descriptive Statistics (6)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Use central tendency measures and dispersion measures to describe the data gathered. • Use graphics to communicate the characteristics of the data gathered.	• Central Tendency (Mean, median, mode) • Dispersion (Range, standard deviation, quartile) • Graphics: histogram, boxplot, etc.: Communication ability.
Readings : [MRo14], [Men14]	

Unit 3: Inferential Statistics (6)	
Competences Expected: CS2	
Learning Outcomes	**Topics**
• Propose questions and hypotheses of interest. • Analyze the data gathered using different statistical tools to answer questions of interest. • Draw conclusions based on the analysis performed.	• Determination of the sample size • Confidence interval • Type I and type II error • Distribution type • Hypothesis test (t-student, means, proportions and ANOVA) • Relationships between variables: correlation, regression.
Readings : [MRo14], [Men14]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** GH2101. English II (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 10 HP; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Cam06] Cambridge. *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford, 2006.

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

6. **Information about the course**

(a) **Brief description about the course** A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of life. In the case of foreign languages, undoubtedly English is the most practical because it is spoken around the world. There is no country where it is not spoken. In careers related to tourist services, English is perhaps the most important practical tool that the student must master from the outset as part of his / her integral education

(b) **Prerrequisites:** GH1102. English II. (3^{rd} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Train the student to understand and hold a conversation.
- Provide techniques of llation of ideas .

8. **Contribution to Outcomes**

f) An ability to communicate effectively. (**Usage**)

9. **Competences (IEEE)**

C25. Ability to communicate in a second language.⇒ **Outcome f**

10. **List of topics**

1. Getting to know you!
2. The way we live!
3. It all went wrong!
4. Let's go shopping!
5. What do you want to do?

6. The best in the world!

7. Fame!

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Getting to know you! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the first unit, each of the students, understanding the grammar of present, past and future times, is able to express a greater number of actions in the form of sentences. He is also able to express ideas in the form of questions. Assume the idea of words with more than one meaning. Use social expressions in entertainment situations.	• Present, Past, and Future Times. • Interrogative sentences with Wh-. • Words with more than one meaning. • Parts of the sentence • Expressions for free time
Readings : [SJ02], [Cam06], [Mac99]	

Unit 2: The way we live! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• Ath the end of the second unit, students having identified the present form of expression recognize the difference between the forms of the same and apply it properly. They describe the countries accurately. They take expressions to show interest. Use connectors to join various ideas.	• Simple present tense. • Present Continuous Time. • Collocations. • Vocabulary of the countries of the world. • Expressions of anger. • Connectors.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 3: It all went wrong! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the third unit, students having recognized the characteristics of past times use them properly. They use prefixes and suffixes to create and recognize new words. They describe time in a broad way. They will use conjunctions to unite type ideas.	• Past simple tense. • Continuous past tense. • Irregular Verbs. • Time expressions. • Connectors of time.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 4: Let's go shopping! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the fourth unit, students having identified the idea of quantity express different situations that involve it. Recognize and apply articles to nouns. They assume the idea of shopping with the help of expressions. They express money prices and ideas. They fill several formats. They express attitudes.	• Expressions of Indefinite Quantity. • Affirmative sentences, Negatives and Questions. • Use of Articles. • Product prices. • Filling of formats and surveys • Expressions for shopping
Readings : [SJ02], [Cam06], [Mac99]	

Unit 5: What do you want to do? (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the fifth unit, students, from the understanding of the idea of verbal patterns, will elaborate sentences using the necessary elements. They will also assimilate the need to express future intentions. They will acquire vocabulary to describe feelings. Expressions will be presented to describe plans and ambitions.	• Verbal Patterns I. • Future Intentions. • Verbs of Perception. • Vocabulary of feelings. • Expressions of Plans and Ambitions.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 6: The best in the world! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the sixth unit, students having known the fundamentals of using adjectives, structure sentences with different forms of adjectives in appropriate contexts. They emphasize the difference between types of cities and towns and lifestyles. They use expressions indicating directions.	• WhatŽs it like?. • Adjectives • Comparative and superlative. • Synonyms and antonyms. • Indications of direction . • Readings.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 7: Fame! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the seventh unit, students have learned the fundamentals of structuring the present perfect time and differentiate it from the simple past. They emphasize the difference between forms of adjectives. Describe ideas of music. They use expressions to give short answers. They assume the idea of giving extra explanations of the elements of a sentence.	• Present Perfect and Simple Past • Expressions for, ever, since • Adverbs • Expressions that come in pairs • Short answers • Celebrities
Readings : [SJ02], [Cam06], [Mac99]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS2102. Analysis and Design of Algorithms (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Als99] H. Alsuwaiyel. *Algorithms: Design Techniques and Analysis.* World Scientific, 1999. ISBN: 9789810237400.

[DPV06] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms.* McGraw-Hill Education, 2006. ISBN: 9780073523408.

[GT09] Michael T. Goodrich and Roberto Tamassia. *Algorithm Design: Foundations, Analysis and Internet Examples.* 2nd. John Wiley & Sons, Inc., 2009. ISBN: 0470088540, 9780470088548.

[Knu97] D.E. Knuth. *The Art of Computer Programming: Fundamental algorithms Vol 1.* Third Edition. Addison-Wesley, 1997. ISBN: 9780201896831. URL: `http://www-cs-faculty.stanford/~knuth/taocp.html`.

[KT05] Jon Kleinberg and Eva Tardos. *Algorithm Design.* Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN: 0321295358.

[Raw92] G.J.E. Rawlins. *Compared to What?: An Introduction to the Analysis of Algorithms.* Computer Science Press, 1992. ISBN: 9780716782438.

[RS09] Thomas H. Cormen; Charles E. Leiserson ; Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms, Third Edition.* 3rd. The MIT Press, 2009. ISBN: 0262033844.

[SF13] R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms.* Pearson Education, 2013. ISBN: 9780133373486.

[SW11] R. Sedgewick and K. Wayne. *Algorithms.* Pearson Education, 2011. ISBN: 9780132762564.

[Tar83] Robert Endre Tarjan. *Data Structures and Network Algorithms.* Society for Industrial and Applied Mathematics, 1983. ISBN: 0-89871-187-8.

6. **Information about the course**

   (a) **Brief description about the course** An algorithm is, essentially, a well-defined set of rules or instructions that allow solving a computational problem. The theoretical study of the performance of the algorithms and the resources used by them, usually time and space, allows us to evaluate if an algorithm is suitable for solving a specific problem, comparing it with other algorithms for the same problem or even delimiting the boundary between Viable and impossible. This matter is so important that even Donald E. Knuth defined Computer Science as the study of algorithms. This course will present the most common techniques used in the analysis and design of efficient algorithms, with the purpose of learning the fundamental principles of the design, implementation and analysis of algorithms for the solution of computational problems

   (b) **Prerrequisites:** CS2100. Algorithms and Data Structures. (4th Sem)

   (c) **Type of Course:** Mandatory

   (d) **Modality:** Face to face

7. **Specific goals of the Course**

- Develop the ability to evaluate the complexity and quality of algorithms proposed for a given problem.

- Study the most representative, introductory algorithms of the most important classes of problems treated in computation.
- Develop the ability to solve algorithmic problems using the fundamental principles of algorithm design learneds.
- Be able to answer the following questions when a new algorithm is presented: How good is the performance ?, Is there a better way to solve the problem?

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome b**

C3. An intellectual understanding of, and an appreciation for, the central role of algorithms and data structures.⇒ **Outcome b**

C5. Ability to implement algorithms and data structures in software.⇒ **Outcome a**

C6. Ability to design and implement larger structural units that utilize algorithms and data structures and the interfaces through which these units communicate.⇒ **Outcome a**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome a**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome b**

C3. An intellectual understanding of, and an appreciation for, the central role of algorithms and data structures.⇒ **Outcome b**

C5. Ability to implement algorithms and data structures in software.⇒ **Outcome a**

C6. Ability to design and implement larger structural units that utilize algorithms and data structures and the interfaces through which these units communicate.⇒ **Outcome a**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome a**

10. List of topics

1. Basic Analysis
2. Algorithmic Strategies
3. Fundamental Data Structures and Algorithms
4. Basic Automata Computability and Complexity
5. Advanced Data Structures Algorithms and Analysis

11. Methodology and Evaluation

Methodology:

Theory Sessions:

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:

In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:

Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Basic Analysis (10)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm [Assessment] • In the context of specific algorithms, identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors [Assessment] • Determine informally the time and space complexity of simple algorithms [Assessment] • State the formal definition of big O [Assessment] • List and contrast standard complexity classes [Assessment] • Use big O notation formally to give asymptotic upper bounds on time and space complexity of algorithms [Assessment] • Use big O notation formally to give expected case bounds on time complexity of algorithms [Assessment] • Explain the use of big omega, big theta, and little o notation to describe the amount of work done by an algorithm [Assessment] • Use recurrence relations to determine the time complexity of recursively defined algorithms [Assessment] • Solve elementary recurrence relations, eg, using some form of a Master Theorem [Assessment]	• Differences among best, expected, and worst case behaviors of an algorithm • Asymptotic analysis of upper and expected complexity bounds • Big O notation: formal definition • Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential • Big O notation: use • Recurrence relations • Analysis of iterative and recursive algorithms • Some version of a Master Theorem
Readings : [KT05], [DPV06], [RS09], [SF13], [Knu97]	

<table>
<tr><td colspan="2"><b>Unit 2: Algorithmic Strategies (30)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: C2</b></td></tr>
<tr><td><b>Learning Outcomes</b></td><td><b>Topics</b></td></tr>
<tr><td>• For each of the strategies (brute-force, greedy, divide-and-conquer, recursive backtracking, and dynamic programming), identify a practical example to which it would apply [Assessment]
• Use a greedy approach to solve an appropriate problem and determine if the greedy rule chosen leads to an optimal solution [Assessment]
• Use a divide-and-conquer algorithm to solve an appropriate problem [Assessment]
• Use dynamic programming to solve an appropriate problem [Assessment]
• Determine an appropriate algorithmic approach to a problem [Assessment]</td><td>• Brute-force algorithms
• Greedy algorithms
• Divide-and-conquer
• Dynamic Programming</td></tr>
<tr><td colspan="2"><b>Readings :</b> [KT05], [DPV06], [RS09], [Als99]</td></tr>
</table>

Unit 3: Fundamental Data Structures and Algorithms (10)	
Competences Expected: C6	
Learning Outcomes	**Topics**
• Implement basic numerical algorithms [Assessment] • Implement simple search algorithms and explain the differences in their time complexities [Assessment] • Be able to implement common quadratic and O(N log N) sorting algorithms [Assessment] • Discuss the runtime and memory efficiency of principal algorithms for sorting, searching, and hashing [Usage] • Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Familiarity] • Solve problems using fundamental graph algorithms, including depth-first and breadth-first search [Assessment] • Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context [Assessment] • Describe the heap property and the use of heaps as an implementation of priority queues [Assessment] • Solve problems using graph algorithms, including single-source and all-pairs shortest paths, and at least one minimum spanning tree algorithm [Assessment]	• Simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max, • Sequential and binary search algorithms • Worst case quadratic sorting algorithms (selection, insertion) • Worst or average case O(N log N) sorting algorithms (quicksort, heapsort, mergesort) • Graphs and graph algorithms – Representations of graphs (e.g., adjacency list, adjacency matrix) – Depth- and breadth-first traversals • Heaps • Graphs and graph algorithms – Shortest-path algorithms (Dijkstra's and Floyd's algorithms) – Minimum spanning tree (Prim's and Kruskal's algorithms)
Readings : [KT05], [DPV06], [RS09], [SW11], [GT09]	

Unit 4: Basic Automata Computability and Complexity (2)	
Competences Expected: C9	
Learning Outcomes	**Topics**
• Define the classes P and NP [Familiarity] • Explain the significance of NP-completeness [Familiarity]	• Introduction to the P and NP classes and the P vs. NP problem • Introduction to the NP-complete class and exemplary NP-complete problems (e.g., SAT, Knapsack)
Readings : [KT05], [DPV06], [RS09]	

Unit 5: Advanced Data Structures Algorithms and Analysis (8)	
Competences Expected: C16	
Learning Outcomes	**Topics**
• Understand the mapping of real-world problems to algorithmic solutions (eg, as graph problems, linear programs, etc) [Familiarity] • Select and apply advanced analysis techniques (eg, amortized, probabilistic, etc) to algorithms [Usage]	• Graphs (e.g, topological sort, finding strongly connected components, matching) • Number-theoretic algorithms (e.g., modular arithmetic, primality testing, integer factorization) • Randomized algorithms • Amortized analysis • Probabilistic analysis
Readings : [KT05], [DPV06], [RS09], [Tar83], [Raw92]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS2702. Databases II (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 1 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Bur04] Donald K. Burleson. *Physical Database Design Using Oracle*. CRC Press, 2004.

[Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

[Dat05] C.J. Date. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Elsevier, 2005.

[M T99] Patrick Valduriez M. Tamer Ozsu. *Principles of Distributed Database Systems, Second Edition*. Prentice Hall, 1999.

[Pet98] Julita Vassileva Peter Brusilovsky Alfred Kobsa. *Adaptive Hypertext and Hypermedia, First Edition*. Springer, 1998.

[Phi97] Eric Newcomer Philip A. Bernstein. *Principles of Transaction Processing, First Edition*. Morgan Kaufmann, 1997.

[Ram04] Shamkant B. Navathe Ramez Elmasri. *Fundamentals of Database Systems, Fourth Edition*. Addison Wesley, 2004.

6. **Information about the course**

(a) **Brief description about the course** Information Management (IM) plays a leading role in almost every area where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of access and update of stored information, data modeling and abstraction, and physical file storage techniques.

It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which IM methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable constraints, including scalability and Usability.

(b) **Prerrequisites:** CS2701. Databases I. (4^{th} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- To make the student understand the different applications that the databases have, in the different areas of knowledge.
- Show appropriate ways of storing information based on their various approaches and their subsequent retrieval of information.

8. **Contribution to Outcomes**

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

k) Apply the principles of development and design in the construction of software systems of variable complexity. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

k) Apply the principles of development and design in the construction of software systems of variable complexity. (**Usage**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome b**

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome j**

CS4. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.⇒ **Outcome k**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome b**

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome j**

CS4. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.⇒ **Outcome k**

10. List of topics

1. Physical Database Design
2. Transaction Processing
3. Information Storage and Retrieval
4. Distributed Databases

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Physical Database Design (10)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Explain the concepts of records, record types, and files, as well as the different techniques for placing file records on disk [Usage] • Give examples of the application of primary, secondary, and clustering indexes [Usage] • Distinguish between a non-dense index and a dense index [Usage] • Implement dynamic multilevel indexes using B-trees [Usage] • Explain the theory and application of internal and external hashing techniques [Usage] • Use hashing to facilitate dynamic file expansion [Usage] • Describe the relationships among hashing, compression, and efficient database searches [Usage] • Evaluate costs and benefits of various hashing schemes [Usage] • Explain how physical database design affects database transaction efficiency [Usage]	• Storage and file structure • Indexed files • Hashed files • Signature files • B-trees • Files with dense index • Files with variable length records • Database efficiency and tuning
Readings : [Bur04], [Dat05], [Cel05]	

Unit 2: Transaction Processing (12)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Create a transaction by embedding SQL into an application program [Usage] • Explain the concept of implicit commits [Usage] • Describe the issues specific to efficient transaction execution [Usage] • Explain when and why rollback is needed and how logging assures proper rollback [Usage] • Explain the effect of different isolation levels on the concurrency control mechanisms [Usage] • Choose the proper isolation level for implementing a specified transaction protocol [Usage] • Identify appropriate transaction boundaries in application programs [Usage]	• Transactions • Failure and recovery • Concurrency control • Interaction of transaction management with storage, especially buffering
Readings : [Phi97], [Ram04]	

Unit 3: Information Storage and Retrieval (10)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Explain basic information storage and retrieval concepts [Usage] • Describe what issues are specific to efficient information retrieval [Usage] • Give applications of alternative search strategies and explain why the particular search strategy is appropriate for the application [Usage] • Design and implement a small to medium size information storage and retrieval system, or digital library [Usage] • Describe some of the technical solutions to the problems related to archiving and preserving information in a digital library [Usage]	• Documents, electronic publishing, markup, and markup languages • Tries, inverted files, PAT trees, signature files, indexing • Morphological analysis, stemming, phrases, stop lists • Term frequency distributions, uncertainty, fuzziness, weighting • Vector space, probabilistic, logical, and advanced models • Information needs, relevance, evaluation, effectiveness • Thesauri, ontologies, classification and categorization, metadata • Bibliographic information, bibliometrics, citations • Routing and (community) filtering • Multimedia search, information seeking behavior, user modeling, feedback • Information summarization and visualization • Faceted search (e.g., using citations, keywords, classification schemes) • Digital libraries • Digitization, storage, interchange, digital objects, composites, and packages • Metadata and cataloging • Naming, repositories, archives • Archiving and preservation, integrity • Spaces (conceptual, geographical, 2/3D, VR) • Architectures (agents, buses, wrappers/mediators), interoperability • Services (searching, linking, browsing, and so forth) • Intellectual property rights management, privacy, and protection (watermarking)
Readings : [Pet98], [Ram04]	

Unit 4: Distributed Databases (36)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Explain the techniques used for data fragmentation, replication, and allocation during the distributed database design process [Usage] • Evaluate simple strategies for executing a distributed query to select the strategy that minimizes the amount of data transfer [Usage] • Explain how the two-phase commit protocol is used to deal with committing a transaction that accesses databases stored on multiple nodes [Usage] • Describe distributed concurrency control based on the distinguished copy techniques and the voting method [Usage] • Describe the three levels of software in the client-server model [Usage]	• Distributed DBMS – Distributed data storage – Distributed query processing – Distributed transaction model – Homogeneous and heterogeneous solutions – Client-server distributed databases • Parallel DBMS – Parallel DBMS architectures: shared memory, shared disk, shared nothing; – Speedup and scale-up, e.g., use of the MapReduce processing model – Data replication and weak consistency models
Readings : [M T99], [Dat05]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: CS2901. Software Engineering I (Mandatory)
2. Credits: 4
3. Hours of theory and Lab: 2 HT; 4 HL; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[Lar08] Craig Larman. *Applying UML and Patterns*. Prentice Hall, 2008.

[Pre05] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. 6th. McGraw-Hill, Mar. 2005.

[Som08] Ian Sommerville. *Software Engineering*. 7th. ISBN: 0321210263. Addison Wesley, May 2008.

6. Information about the course

(a) **Brief description about the course** The aim of developing software, except for extremely simple applications, requires the execution of a well-defined development process. Professionals in this area require a high degree of knowledge of the different models and development process, so that they are able to choose the most suitable for each development project. On the other hand, the development of medium and large-scale systems requires the use of pattern and component libraries and the mastery of techniques related to component-based design

(b) **Prerrequisites:**

- CS1103. Objects oriented programming II. (3^{rd} Sem)
- CS2701. Databases I. (4^{th} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Provide the student with a theoretical and practical framework for the development of software under quality standards.
- Familiarize the student with the software modeling and construction processes through the use of CASE tools.
- Students should be able to select architectures and ad-hoc technology platforms for deployment scenarios
- Applying component-based modeling to ensure variables such as quality, cost, and time-to-market in development processes.
- Provide students with best practices for software verification and validation.

8. Contribution to Outcomes

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

k) Apply the principles of development and design in the construction of software systems of variable complexity. (**Usage**)

9. Competences (IEEE)

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome b,k**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome b,c,k**

C12. Understanding the lifecycle implications for the development of all aspects of computer-related systems (including software, hardware, and human computer interface).⇒ **Outcome c,i**

C18. Ability to participate actively and as a member of a team. .⇒ **Outcome k**

CS1. Model and design computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.⇒ **Outcome c**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome b,c**

CS4. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.⇒ **Outcome b,c,i**

CS5. Specify, design, and implement computer-based systems.⇒ **Outcome b,c,i**

CS10. Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems. This should include tools for software control including version control and configuration management.⇒ **Outcome i,k**

10. List of topics

1. Requirements Engineering
2. Software Design
3. Software Construction

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Requirements Engineering (18)	
Competences Expected: C7, C11, CS2	
Learning Outcomes	**Topics**
• List the key components of a use case or similar description of some behavior that is required for a system [Assessment] • Describe how the requirements engineering process supports the elicitation and validation of behavioral requirements [Assessment] • Interpret a given requirements model for a simple software system [Assessment] • Describe the fundamental challenges of and common techniques used for requirements elicitation [Assessment] • List the key components of a data model (eg, class diagrams or ER diagrams) [Assessment] • Identify both functional and non-functional requirements in a given requirements specification for a software system [Assessment] • Conduct a review of a set of software requirements to determine the quality of the requirements with respect to the characteristics of good requirements [Assessment] • Apply key elements and common methods for elicitation and analysis to produce a set of software requirements for a medium-sized software system [Assessment] • Compare the plan-driven and agile approaches to requirements specification and validation and describe the benefits and risks associated with each [Assessment] • Use a common, non-formal method to model and specify the requirements for a medium-size software system [Assessment] • Translate into natural language a software requirements specification (eg, a software component contract) written in a formal specification language [Assessment] • Create a prototype of a software system to mitigate risk in requirements [Assessment] • Differentiate between forward and backward tracing and explain their roles in the requirements validation process [Assessment]	• Describing functional requirements using, for example, use cases or users stories • Properties of requirements including consistency, validity, completeness, and feasibility • Software requirements elicitation • Describing system data using, for example, class diagrams or entity-relationship diagrams • Non functional requirements and their relationship to software quality • Evaluation and use of requirements specifications • Requirements analysis modeling techniques • Acceptability of certainty / uncertainty considerations regarding software / system behavior • Prototyping • Basic concepts of formal requirements specification • Requirements specification • Requirements validation • Requirements tracing
Readings : [Pre05], [Som08], [Lar08]	

Unit 2: Software Design (18)	
Competences Expected: C5, C7, C8, CS10	
Learning Outcomes	**Topics**
• Articulate design principles including separation of concerns, information hiding, coupling and cohesion, and encapsulation [Familiarity] • Use a design paradigm to design a simple software system, and explain how system design principles have been applied in this design [Usage] • Construct models of the design of a simple software system that are appropriate for the paradigm used to design it [Usage] • Within the context of a single design paradigm, describe one or more design patterns that could be applicable to the design of a simple software system [Familiarity] • For a simple system suitable for a given scenario, discuss and select an appropriate design paradigm [Usage] • Create appropriate models for the structure and behavior of software products from their requirements specifications [Usage] • Explain the relationships between the requirements for a software product and its design, using appropriate models [Assessment] • For the design of a simple software system within the context of a single design paradigm, describe the software architecture of that system [Familiarity] • Given a high-level design, identify the software architecture by differentiating among common software architectures such as 3-tier, pipe-and-filter, and client-server [Familiarity] • Investigate the impact of software architectures selection on the design of a simple system [Assessment] • Apply simple examples of patterns in a software design [Usage] • Describe a form of refactoring and discuss when it may be applicable [Familiarity] • Select suitable components for use in the design of a software product [Usage] • Explain how suitable components might need to be adapted for use in the design of a software product [Familiarity] • Design a contract for a typical small software component for use in a given system [Usage] • Discuss and select appropriate software architecture for a simple system suitable for a given scenario [Usage] • Apply models for internal and external qualities in designing software components to achieve an acceptable tradeoff between conflicting quality aspects [Us	• System design principles: levels of abstraction (architectural design and detailed design), separation of concerns, information hiding, coupling and cohesion , re-use of standard structures • Design Paradigms such as structured design (top-down functional decomposition), object-oriented analysis and design, event driven design, component-level design, data-structured centered, aspect oriented, function oriented, service oriented • Structural and behavioral models of software designs • Design patterns • Relationships between requirements and designs: transformation of models, design of contracts, invariants • Software architecture concepts and standard architectures (e.g. client-server, n-layer, transform centered, pipes-and-filters) • The use of component desing: component selection, design, adaptation and assembly of components, component and patterns, components and objects (for example, building a GUI using a standar widget set) • Refactoring designs using design patterns • Internal design qualities, and models for them: efficiency and performance, redundacy and fault tolerance, traceability of requeriments • Measurement and analysis of design quality • Tradeoffs between different aspects of quality • Application frameworks • Middleware: the object-oriented paradigm within middleware, object request brokers and marshalling, transaction processing monitors, workflow systems • Principles of secure design and coding – Principle of least privilege – Principle of fail-safe defaults – Principle of psychological acceptability

Unit 3: Software Construction (24)	
Competences Expected: C4, C5, C7, C8, CS2	
Learning Outcomes	**Topics**
• Describe techniques, coding idioms and mechanisms for implementing designs to achieve desired properties such as reliability, efficiency, and robustness [Assessment] • Build robust code using exception handling mechanisms [Assessment] • Describe secure coding and defensive coding practices [Assessment] • Select and use a defined coding standard in a small software project [Assessment] • Compare and contrast integration strategies including top-down, bottom-up, and sandwich integration [Assessment] • Describe the process of analyzing and implementing changes to code base developed for a specific project [Assessment] • Describe the process of analyzing and implementing changes to a large existing code base [Assessment] • Rewrite a simple program to remove common vulnerabilities, such as buffer overflows, integer overflows and race conditions [Assessment] • Write a software component that performs some non-trivial task and is resilient to input and run-time errors [Assessment]	• Coding practices: techniques, idioms/patterns, mechanisms for building quality programs – Defensive coding practices – Secure coding practices – Using exception handling mechanisms to make programs more robust, fault-tolerant • Coding standards • Integration strategies • Development context: “green field” vs. existing code base – Change impact analysis – Change actualization • Potential security problems in programs – Buffer and other types of overflows – Race conditions – Improper initialization, including choice of privileges – Checking input – Assuming success and correctness – Validating assumptions
Readings : [Pre05], [Som08], [Lar08]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: CS2S01. Operating systems (Mandatory)
2. Credits: 4
3. Hours of theory and Lab: 2 HT; 4 HL; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[Avi12] Greg Gagne Avi Silberschatz Peter Baer Galvin. *Operating System Concepts, 9/E.* John Wiley & Sons, Inc., 2012. ISBN: 978-1-118-06333-0.

[Mat99] Luis Mateu. *Apuntes de Sistemas Operativos.* Universidad de Chile, 1999.

[Sta05] William Stallings. *Operating Systems: Internals and Design Principles, 5/E.* Prentice Hall, 2005. ISBN: 0-13-147954-7.

[Tan01] Andrew S. Tanenbaum. *Modern Operating Systems, 2/E.* Prentice Hall, 2001. ISBN: 0-13-031358-0.

[Tan06] Andrew S. Tanenbaum. *Operating Systems Design and Implementation, 3/E.* Prentice Hall, 2006. ISBN: 0-13-142938-8.

6. Information about the course

(a) **Brief description about the course** An Operating System is a program that acts as an intermediary between the user and the machine.

The purpose of an operating system is to provide an environment in which the user can run their applications.

In this course the design of the core of the operating systems will be studied. In addition, the course includes practical activities in which problems of concurrency will be solved and the operation of a pseudo Operating System will be modified.

(b) **Prerrequisites:** CS2201. Computer Architecture. (3^{rd} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Know the basic elements of the design of the operating systems.

8. Contribution to Outcomes

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome b**

C6. Ability to design and implement larger structural units that utilize algorithms and data structures and the interfaces through which these units communicate.⇒ **Outcome b**

CS8. Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, multimedia systems and mobile systems..⇒ **Outcome b**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome b**

C6. Ability to design and implement larger structural units that utilize algorithms and data structures and the interfaces through which these units communicate.⇒ **Outcome b**

CS8. Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, multimedia systems and mobile systems..⇒ **Outcome b**

10. List of topics

1. Overview of Operating Systems
2. Operating System Principles
3. Concurrency
4. Scheduling and Dispatch
5. Memory Management
6. Security and Protection
7. Virtual Machines
8. Device Management
9. File Systems
10. Real Time and Embedded Systems
11. Fault Tolerance
12. System Performance Evaluation

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Overview of Operating Systems (3)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Explain the objectives and functions of modern operating systems [Familiarity] • Analyze the tradeoffs inherent in operating system design [Assessment] • Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve [Familiarity] • Discuss networked, client-server, distributed operating systems and how they differ from single user operating systems [Familiarity] • Identify potential threats to operating systems and the security features design to guard against them [Familiarity]	• Role and purpose of the operating system • Functionality of a typical operating system • Mechanisms to support client-server models, handheld devices • Design issues (efficiency, robustness, flexibility, portability, security, compatibility) • Influences of security, networking, multimedia, windowing systems
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 2: Operating System Principles (6)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Explain the concept of a logical layer [Familiarity] • Explain the benefits of building abstract layers in hierarchical fashion [Familiarity] • Describe the value of APIs and middleware [Familiarity] • Describe how computing resources are used by application software and managed by system software [Familiarity] • Contrast kernel and user mode in an operating system [Assessment] • Discuss the advantages and disadvantages of using interrupt processing [Familiarity] • Explain the use of a device list and driver I/O queue [Familiarity]	• Structuring methods (monolithic, layered, modular, micro-kernel models) • Abstractions, processes, and resources • Concepts of application program interfaces (APIs) • The evolution of hardware/software techniques and application needs • Device organization • Interrupts: methods and implementations • Concept of user/system state and protection, transition to kernel mode
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 3: Concurrency (9)	
Competences Expected: C6	
Learning Outcomes	**Topics**
• Describe the need for concurrency within the framework of an operating system [Familiarity] • Demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks [Usage] • Summarize the range of mechanisms that can be employed at the operating system level to realize concurrent systems and describe the benefits of each [Familiarity] • Explain the different states that a task may pass through and the data structures needed to support the management of many tasks [Familiarity] • Summarize techniques for achieving synchronization in an operating system (eg, describe how to implement a semaphore using OS primitives) [Familiarity] • Describe reasons for using interrupts, dispatching, and context switching to support concurrency in an operating system [Familiarity] • Create state and transition diagrams for simple problem domains [Usage]	• States and state diagrams • Structures (ready list, process control blocks, and so forth) • Dispatching and context switching • The role of interrupts • Managing atomic access to OS objects • Implementing synchronization primitives • Multiprocessor issues (spin-locks, reentrancy)
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

<table>
<tr><td colspan="2"><b>Unit 4: Scheduling and Dispatch (6)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: CS8</b></td></tr>
<tr><td><b>Learning Outcomes</b></td><td><b>Topics</b></td></tr>
<tr><td>• Compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems, such as priority, performance comparison, and fair-share schemes [Assessment]
• Describe relationships between scheduling algorithms and application domains [Familiarity]
• Discuss the types of processor scheduling such as short-term, medium-term, long-term, and I/O [Familiarity]
• Describe the difference between processes and threads [Familiarity]
• Compare and contrast static and dynamic approaches to real-time scheduling [Assessment]
• Discuss the need for preemption and deadline scheduling [Familiarity]
• Identify ways that the logic embodied in scheduling algorithms are applicable to other domains, such as disk I/O, network scheduling, project scheduling, and problems beyond computing [Familiarity]</td><td>• Preemptive and non-preemptive scheduling
• Schedulers and policies
• Processes and threads
• Deadlines and real-time issues</td></tr>
<tr><td colspan="2"><b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Unit 5: Memory Management (6)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: C1</b></td></tr>
<tr><td><b>Learning Outcomes</b></td><td><b>Topics</b></td></tr>
<tr><td>• Explain memory hierarchy and cost-performance trade-offs [Familiarity]
• Summarize the principles of virtual memory as applied to caching and paging [Familiarity]
• Evaluate the trade-offs in terms of memory size (main memory, cache memory, auxiliary memory) and processor speed [Assessment]
• Defend the different ways of allocating memory to tasks, citing the relative merits of each [Familiarity]
• Describe the reason for and use of cache memory (performance and proximity, different dimension of how caches complicate isolation and VM abstraction) [Familiarity]
• Discuss the concept of thrashing, both in terms of the reasons it occurs and the techniques used to recognize and manage the problem [Familiarity]</td><td>• Review of physical memory and memory management hardware
• Working sets and thrashing
• Caching</td></tr>
<tr><td colspan="2"><b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]</td></tr>
</table>

Unit 6: Security and Protection (6)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Articulate the need for protection and security in an OS [Familiarity] • Summarize the features and limitations of an operating system used to provide protection and security [Familiarity] • Explain the mechanisms available in an OS to control access to resources (cross reference IAS/Security Architecture and Systems Administration/Access Control/Configuring systems to operate securely as an IT system) [Familiarity] • Carry out simple system administration tasks according to a security policy, for example creating accounts, setting permissions, applying patches, and arranging for regular backups (cross reference IAS/Security Architecture and Systems Administration) [Familiarity]	• Overview of system security • Policy/mechanism separation • Security methods and devices • Protection, access control, and authentication • Backups
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 7: Virtual Machines (6)	
Competences Expected: CS8	
Learning Outcomes	**Topics**
• Explain the concept of virtual memory and how it is realized in hardware and software [Familiarity] • Differentiate emulation and isolation [Familiarity] • Evaluate virtualization trade-offs [Assessment] • Discuss hypervisors and the need for them in conjunction with different types of hypervisors [Familiarity]	• Types of virtualization (including Hardware/Software, OS, Server, Service, Network) • Paging and virtual memory • Virtual file systems • Hypervisors • Portable virtualization; emulation vs. isolation • Cost of virtualization
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 8: Device Management (6)	
Competences Expected: C6	
Learning Outcomes	**Topics**
• Explain the key difference between serial and parallel devices and identify the conditions in which each is appropriate [Familiarity] • Identify the relationship between the physical hardware and the virtual devices maintained by the operating system [Familiarity] • Explain buffering and describe strategies for implementing it [Familiarity] • Differentiate the mechanisms used in interfacing a range of devices (including hand-held devices, networks, multimedia) to a computer and explain the implications of these for the design of an operating system [Familiarity] • Describe the advantages and disadvantages of direct memory access and discuss the circumstances in which its use is warranted [Familiarity] • Identify the requirements for failure recovery [Familiarity] • Implement a simple device driver for a range of possible devices [Usage]	• Characteristics of serial and parallel devices • Abstracting device differences • Buffering strategies • Direct memory access • Recovery from failures
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unit 9: File Systems (6)	
Competences Expected: CS8	
Learning Outcomes	**Topics**
• Describe the choices to be made in designing file systems [Familiarity] • Compare and contrast different approaches to file organization, recognizing the strengths and weaknesses of each [Assessment] • Summarize how hardware developments have led to changes in the priorities for the design and the management of file systems [Familiarity] • Summarize the use of journaling and how log-structured file systems enhance fault tolerance [Familiarity]	• Files: data, metadata, operations, organization, buffering, sequential, nonsequential. • Directories: contents and structure. • File systems: partitioning, mount/unmount, virtual file systems. • Standard implementation techniques • Memory-mapped files • Special-purpose file systems. • Naming, searching, access, backups • Journaling and log-structured file systems
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

<table>
<tr><th colspan="2">Unit 10: Real Time and Embedded Systems (6)</th></tr>
<tr><th colspan="2">Competences Expected: C1</th></tr>
<tr><th>Learning Outcomes</th><th>Topics</th></tr>
<tr><td>• Describe what makes a system a real-time system [Familiarity]
• Explain the presence of and describe the characteristics of latency in real-time systems [Familiarity]
• Summarize special concerns that real-time systems present, including risk, and how these concerns are addressed [Familiarity]</td><td>• Process and task scheduling
• Memory/disk management requirements in a real-time environment
• Failures, risks, and recovery.
• Special concerns in real-time systems</td></tr>
<tr><td colspan="2"><b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]</td></tr>
</table>

<table>
<tr><th colspan="2">Unit 11: Fault Tolerance (3)</th></tr>
<tr><th colspan="2">Competences Expected: C1</th></tr>
<tr><th>Learning Outcomes</th><th>Topics</th></tr>
<tr><td>• Explain the relevance of the terms fault tolerance, reliability, and availability [Familiarity]
• Outline the range of methods for implementing fault tolerance in an operating system [Familiarity]
• Explain how an operating system can continue functioning after a fault occurs [Familiarity]</td><td>• Fundamental concepts: reliable and available systems
• Spatial and temporal redundancy
• Methods used to implement fault tolerance
• Examples of OS mechanisms for detection, recovery, restart to implement fault tolerance, use of these techniques for the OS's own services.</td></tr>
<tr><td colspan="2"><b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]</td></tr>
</table>

<table>
<tr><th colspan="2">Unit 12: System Performance Evaluation (3)</th></tr>
<tr><th colspan="2">Competences Expected: C1</th></tr>
<tr><th>Learning Outcomes</th><th>Topics</th></tr>
<tr><td>• Describe the performance measurements used to determine how a system performs [Familiarity]
• Explain the main evaluation models used to evaluate a system [Familiarity]</td><td>• Why system performance needs to be evaluated?
• What is to be evaluated?
• Systems performance policies, e.g., caching, paging, scheduling, memory management, and security
• Evaluation models: deterministic, analytic, simulation, or implementation-specific
• How to collect evaluation data (profiling and tracing mechanisms)</td></tr>
<tr><td colspan="2"><b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]</td></tr>
</table>

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS3402. Compilers (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Aho+08] Alfred Aho et al. *Compiladores. Principios, técnicas y herramientas.* 2nd. ISBN:10-970-26-1133-4. Addison Wesley, 2008.

[Aho90] Alfred Aho. *Compiladores Principios, técnicas y herramientas.* Addison Wesley, 1990.

[ALe96] Karen A.Lemone. *Fundamentos de Compiladores.* CECSA-Mexico, 1996.

[App02] A. W. Appel. *Modern compiler implementation in Java.* 2.a edición. Cambridge University Press, 2002.

[Lou04a] Kenneth C. Louden. *Construccion de Compiladores Principios y Practica.* Thomson, 2004.

[Lou04b] Kenneth C. Louden. *Lenguajes de Programacion.* Thomson, 2004.

[PV98] Terrence W. Pratt and Marvin V.Zelkowitz. *Lenguajes de Programacion Diseño e Implementacion.* Prentice-Hall Hispanoamericana S.A., 1998.

[TS98] Bernard Teufel and Stephanie Schmidt. *Fundamentos de Compiladores.* Addison Wesley Iberoamericana, 1998.

6. **Information about the course**

   (a) **Brief description about the course** That the student knows and understands the concepts and fundamental principles of the theory of compilation to realize the construction of a compiler

   (b) **Prerrequisites:** CS2101. Theory of Computation. (4th Sem)

   (c) **Type of Course:** Mandatory

   (d) **Modality:** Face to face

7. **Specific goals of the Course**

- Know the basic techniques used during the process of intermediate generation, optimization and code generation.
- Learning to implement small compilers.

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

9. Competences (IEEE)

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome a**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome b,j**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome a**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome b,j**

10. List of topics

1. Program Representation
2. Language Translation and Execution
3. Syntax Analysis
4. Compiler Semantic Analysis
5. Code Generation

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Program Representation (5)	
Competences Expected: C9	
Learning Outcomes	**Topics**
• Explain how programs that process other programs treat the other programs as their input data [Familiarity] • Describe an abstract syntax tree for a small language [Familiarity] • Describe the benefits of having program representations other than strings of source code [Familiarity] • Write a program to process some representation of code for some purpose, such as an interpreter, an expression optimizer, or a documentation generator [Familiarity] • Explain the use of metadata in run-time representations of objects and activation records, such as class pointers, array lengths, return addresses, and frame pointers [Familiarity] • Discuss advantages, disadvantages, and difficulties of just-in-time and dynamic recompilation [Familiarity] • Identify the services provided by modern language run-time systems [Familiarity]	• Programs that take (other) programs as input such as interpreters, compilers, type-checkers, documentation generators • Abstract syntax trees; contrast with concrete syntax • Data structures to represent code for execution, translation, or transmission • Just-in-time compilation and dynamic recompilation • Other common features of virtual machines, such as class loading, threads, and security.
Readings : [Lou04b], [PV98]	

Unit 2: Language Translation and Execution (10)	
Competences Expected: C8	
Learning Outcomes	**Topics**
• Distinguish a language definition (what constructs mean) from a particular language implementation (compiler vs interpreter, run-time representation of data objects, etc) [Assessment] • Distinguish syntax and parsing from semantics and evaluation [Assessment] • Sketch a low-level run-time representation of core language constructs, such as objects or closures [Assessment] • Explain how programming language implementations typically organize memory into global data, text, heap, and stack sections and how features such as recursion and memory management map to this memory model [Assessment] • Identify and fix memory leaks and dangling-pointer dereferences [Assessment] • Discuss the benefits and limitations of garbage collection, including the notion of reachability [Assessment]	• Interpretation vs. compilation to native code vs. compilation to portable intermediate representation • Language translation pipeline: parsing, optional type-checking, translation, linking, execution – Execution as native code or within a virtual machine – Alternatives like dynamic loading and dynamic (or “just-in-time”) code generation • Run-time representation of core language constructs such as objects (method tables) and first-class functions (closures) • Run-time layout of memory: call-stack, heap, static data – Implementing loops, recursion, and tail calls • Memory management – Manual memory management: allocating, deallocating, and reusing heap memory – Automated memory management: garbage collection as an automated technique using the notion of reachability
Readings : [Aho+08], [Aho90], [Lou04a], [TS98], [ALe96], [App02]	

Unit 3: Syntax Analysis (10)	
Competences Expected: C8	
Learning Outcomes	**Topics**
• Use formal grammars to specify the syntax of languages [Assessment] • Use declarative tools to generate parsers and scanners [Assessment] • Identify key issues in syntax definitions: ambiguity, associativity, precedence [Assessment]	• Scanning (lexical analysis) using regular expressions • Parsing strategies including top-down (e.g., recursive descent, Earley parsing, or LL) and bottom-up (e.g., backtracking or LR) techniques; role of context-free grammars • Generating scanners and parsers from declarative specifications
Readings : [Aho+08], [Aho90], [Lou04a], [TS98], [ALe96], [App02]	

Unit 4: Compiler Semantic Analysis (15)	
Competences Expected: C8	
Learning Outcomes	**Topics**
• Implement context-sensitive, source-level static analyses such as type-checkers or resolving identifiers to identify their binding occurrences [Assessment] • Describe semantic analyses using an attribute grammar [Assessment]	• High-level program representations such as abstract syntax trees • Scope and binding resolution • Type checking • Declarative specifications such as attribute grammars
Readings : [Aho+08], [Aho90], [Lou04a], [TS98], [ALe96], [App02]	

Unit 5: Code Generation (20)	
Competences Expected: C8	
Learning Outcomes	**Topics**
• Identify all essential steps for automatically converting source code into assembly or other low-level languages [Assessment] • Generate the low-level code for calling functions/methods in modern languages [Assessment] • Discuss why separate compilation requires uniform calling conventions [Assessment] • Discuss why separate compilation limits optimization because of unknown effects of calls [Assessment] • Discuss opportunities for optimization introduced by naive translation and approaches for achieving optimization, such as instruction selection, instruction scheduling, register allocation, and peephole optimization [Assessment]	• Procedure calls and method dispatching • Separate compilation; linking • Instruction selection • Instruction scheduling • Register allocation • Peephole optimization
Readings : [Aho+08], [Aho90], [Lou04a], [TS98], [ALe96], [App02]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** GH0010. Ethics and Technology (Mandatory)
2. **Credits:** 2
3. **Hours of theory and Lab:** 1 HT; 2 HP; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Alo06] Garcia. Alonso. *Ética o Filosofía moral*. México,Editorial Trillas, 2006.

[Mar05] Alvarado. Martín. *Ética*. México,Editorial Trillas, 2005.

6. **Information about the course**

(a) **Brief description about the course** This course seeks to provide students with certain frameworks with which to analyze the dilemmas that can be presented in their professional practice. The course puts in practice the critical and responsible reasoning of the students, being this a fundamental competence for the decision-making processes that we will assume as professionals and citizens.

(b) **Prerrequisites:** GH0006. Communication Laboratory II. (2^{nd} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Introduce students to critical and ethical thinking applied to their professional field.
- Developing the competence to look at a phenomenon from various disciplines and perspectives generates in the person empathy and respect for diversity of opinion.
- Ability to work in a team.
- Ability to identify problems
- Oral communication skills
- He is interested in learning about current issues of Peruvian society and the world.
- Written communication skills..

8. **Contribution to Outcomes**

d) An ability to function on multidisciplinary teams. (**Usage**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Usage**)

f) An ability to communicate effectively. (**Usage**)

n) Apply knowledge of the humanities in their professional work. (**Usage**)

o) Improve the conditions of society by putting technology at the service of the human being. (**Usage**)

9. Competences (IEEE)

C10. Understanding of the impact on individuals, organizations, and society of deploying technological solutions and interventions.⇒ **Outcome d,n,o**

C17. Ability to properly express in oral and written media as expected from a university graduate. ⇒ **Outcome f**

C18. Ability to participate actively and as a member of a team. .⇒ **Outcome f**

C21. Understanding the professional, legal, security, political, humanistic, environmental, cultural and ethical issues. ⇒ **Outcome e**

10. List of topics

1. Ética, ciencia y tecnología.
2. Responsabilidad en la ciencia e ingeniería
3. Ciudadanía y ejercicio de la justicia en la era digital

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Ética, ciencia y tecnología. (12)	
Competences Expected: 4	
Learning Outcomes	**Topics**
• Strengthen the student's ability to think interdisciplinarily.	• Definition and scope of ethics Critical thinking / ethical argumentation. • Science and Technology, are engineering and technology issues objective? • Technology: concept and limits. • Importance of ethics in science and engineering.
Readings : [Alo06]	

Unit 2: Responsabilidad en la ciencia e ingeniería (24)	
Competences Expected: 3	
Learning Outcomes	**Topics**
• Understand professional and ethical responsibilities.	• Scope of the concept Responsibility in science (Imperative of Responsability) • Introduction to the subject Responsibility / freedom
Readings : [Mar05]	

Unit 3: Ciudadanía y ejercicio de la justicia en la era digital (30)	
Competences Expected: 3	
Learning Outcomes	**Topics**
• Understands the impact of engineering solutions in a global, economic, environmental and societal context.	• Introduction to the issue of citizenship in the digital age • Technology, new activism and citizenship
Readings : [Mar05]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** ID104. English IV (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 10 HP; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. Bibliography

[Cam06] Cambridge. *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford, 2006.

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

6. Information about the course

(a) **Brief description about the course** A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of life. In the case of foreign languages, English is undoubtedly the most practical because it is spoken around all the world. There is no country where it is not spoken. In addition to being vital to your professional career

(b) **Prerrequisites:** GH2101. English II. (4th Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Increase the level of conversation in different subjects, in the students. As well as the ability to write and read documentation of all kinds.
- Bring the student to a more intense expression in the language domain.

8. Contribution to Outcomes

f) An ability to communicate effectively. (**Usage**)

9. Competences (IEEE)

C25. Ability to communicate in a second language.⇒ **Outcome f**

10. List of topics

1. Do and don't!
2. Going places!
3. Scared to death!
4. Things that changed the world!
5. Dreams and reality!

6. Making a living!
7. All you need is love!

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Do and don't! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the eighth unit, each of the students, understanding the grammar of should and must auxiliaries, is able to express a greater number of actions in an obligatory and suggestive way. Also be able to express ideas describing occupations. Assumes the need to write formal letters	• Mode Auxiliaries should, must and have got to. • Affirmative, negative and interrogative sentences with modals. • Terms for formal letters. • Parts of short answers. • Expressions for occupations.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 2: Going places! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the ninth unit, students having identified how to express present recognize the difference between future forms and apply them properly. They describe conditions accurately. They assume expressions to show place location. They use expressions of time and connectors to unite several ideas.	• Present and Future Present Time with Will • First conditional • Collocations • Vocabulary of prepositions of place and time • Expressions of connection of ideas
Readings : [SJ02], [Cam06], [Mac99]	

Unit 3: Scared to death! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the tenth unit of students, the chapters recognize and use the patterns of times in the past properly. They use exclamation marks. And describe feelings. They will use conjunctions to unite type ideas.	• Infinitive and gerund verb patterns • What + Infinitive • Something + infinitive • Expressions of feelings • Exclamations of surprise
Readings : [SJ02], [Cam06], [Mac99]	

Unit 4: Things that changed the world! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the eleventh unit ,the students having identified the idea of passive actions describe actions appropriately in diverse situations that involve it. They recognize and apply participations. They assume the idea of respecting public signs and signals. They express ideas of habits. They make summaries.	• Passive Voice • Affirmative Prayers, Negatives and Questions • Use of participles, verbs and nouns that go together • Signals. Signs and notes • Summaries • Expressions to indicate prohibition
Readings : [SJ02], [Cam06], [Mac99]	

Unit 5: Dreams and reality! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the twelfth unit, students, starting from understanding the idea of Conditionals and expressing the possibility of elaborating sentences using the necessary elements. They will also assimilate the need for verbal phrases (2 word verbs). They will acquire vocabulary to describe social expressions.	• Second Conditional • Auxiliar of mode ”might” • Phrase Verbs • Social expressions vocabulary • Adverbs • Expressions to give advice
Readings : [SJ02], [Cam06], [Mac99]	

Unit 6: Making a living! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the thirteenth unit, they structure sentences with actions that include present and past in appropriate contexts. They emphasize the difference between types of occupations. Use appropriate expressions for telephone conversations.	• Present Perfect Continuous • Present Continuous • Occupations • Word formation • Adverbs • Expressions of use on the phone
Readings : [SJ02], [Cam06], [Mac99]	

Unit 7: All you need is love! (0)	
Competences Expected: C25	
Learning Outcomes	**Topics**
• At the end of the fourteenth unit, students having learned the fundamentals of structuring past perfect time, differentiate it from the simple past. They emphasize the difference between words in different contexts. Describe farewell ideas. They use expressions to write love stories. They assume the idea of giving and doing interviews.	• Past Perfect and Past Simple • Report Expressions • Expressions of words in different contexts • Short and formal farewells • Love Stories
Readings : [SJ02], [Cam06], [Mac99]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS2301. Networking and Communication (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 1 HT; 4 HL; (15 weeks)
4. **Professor(s)**

   Meetings after coordination with the professor

5. **Bibliography**

[KR13] J.F. Kurose and K.W. Ross. *Computer Networking: A Top-down Approach*. Always learning. Pearson, 2013. ISBN: 9780132856201.

6. **Information about the course**

   (a) **Brief description about the course** The ever-growing development of communication and information technologies means that there is a marked tendency to establish more computer networks that allow better information management..

   In this second course, participants will be introduced to the problems of communication between computers, through the study and implementation of communication protocols such as TCP / IP and the implementation of software on these protocols

   (b) **Prerrequisites:** CS2S01. Operating systems . (5th Sem)

   (c) **Type of Course:** Mandatory

   (d) **Modality:** Face to face

7. **Specific goals of the Course**

   - That the student implements and / or modifies a data communication protocols.
   - That the student master the data transmission techniques used by the existing network protocols.

8. **Contribution to Outcomes**

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Familiarity**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Familiarity**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

9. **Competences (IEEE)**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome c**

C6. Ability to design and implement larger structural units that utilize algorithms and data structures and the interfaces through which these units communicate.⇒ **Outcome c,b**

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome c**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome b**

CS5. Specify, design, and implement computer-based systems.⇒ **Outcome c**

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome c**

CS12. Operate computing equipment and software systems effectively.⇒ **Outcome i**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome c**

C6. Ability to design and implement larger structural units that utilize algorithms and data structures and the interfaces through which these units communicate.⇒ **Outcome c,b**

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome c**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome b**

CS5. Specify, design, and implement computer-based systems.⇒ **Outcome c**

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome c**

CS12. Operate computing equipment and software systems effectively.⇒ **Outcome i**

10. List of topics

1. Introduction
2. Networked Applications
3. Reliable Data Delivery
4. Routing and Forwarding
5. Local Area Networks
6. Resource Allocation
7. Mobility
8. Social Networking

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

Unit 1: Introduction (5)	
Competences Expected: C1,CS8	
Learning Outcomes	**Topics**
• Articulate the organization of the Internet [Familiarity] • List and define the appropriate network terminology [Familiarity] • Describe the layered structure of a typical networked architecture [Familiarity] • Identify the different types of complexity in a network (edges, core, etc) [Familiarity]	• Organization of the Internet (Internet Service Providers, Content Providers, etc.) • Switching techniques (e.g., circuit, packet) • Physical pieces of a network, including hosts, routers, switches, ISPs, wireless, LAN, access point, and firewalls • Layering principles (encapsulation, multiplexing) • Roles of the different layers (application, transport, network, datalink, physical)
Readings : [KR13]	

Unit 2: Networked Applications (5)	
Competences Expected: CS2,CS5	
Learning Outcomes	**Topics**
• List the differences and the relations between names and addresses in a network [Familiarity] • Define the principles behind naming schemes and resource location [Familiarity] • Implement a simple client-server socket-based application [Usage]	• Naming and address schemes (DNS, IP addresses, Uniform Resource Identifiers, etc.) • Distributed applications (client/server, peer-to-peer, cloud, etc.) • HTTP as an application layer protocol • Multiplexing with TCP and UDP • Socket APIs
Readings : [KR13]	

Unit 3: Reliable Data Delivery (10)	
Competences Expected: C6,CS2,CS5	
Learning Outcomes	**Topics**
• Describe the operation of reliable delivery protocols [Familiarity] • List the factors that affect the performance of reliable delivery protocols [Familiarity] • Design and implement a simple reliable protocol [Usage]	• Error control (retransmission techniques, timers) • Flow control (acknowledgements, sliding window) • Performance issues (pipelining) • TCP
Readings : [KR13]	

Unit 4: Routing and Forwarding (12)	
Competences Expected: CS2,CS5	
Learning Outcomes	**Topics**
• Describe the organization of the network layer [Familiarity] • Describe how packets are forwarded in an IP network [Familiarity] • List the scalability benefits of hierarchical addressing [Familiarity]	• Routing versus forwarding • Static routing • Internet Protocol (IP) • Scalability issues (hierarchical addressing)
Readings : [KR13]	

Unit 5: Local Area Networks (10)	
Competences Expected: C1,C7	
Learning Outcomes	**Topics**
• Describe how frames are forwarded in an Ethernet network [Familiarity] • Describe the interrelations between IP and Ethernet [Familiarity] • Describe the interrelations between IP and Ethernet [Familiarity] • Describe the steps used in one common approach to the multiple access problem [Familiarity]	• Multiple Access Problem • Common approaches to multiple access (exponential-backoff, time division multiplexing, etc) • Local Area Networks • Ethernet • Switching
Readings : [KR13]	

Unit 6: Resource Allocation (12)	
Competences Expected: C6,CS5,CS12	
Learning Outcomes	**Topics**
• Describe how resources can be allocated in a network [Familiarity] • Describe the congestion problem in a large network [Familiarity] • Compare and contrast fixed and dynamic allocation techniques [Familiarity] • Compare and contrast current approaches to congestion [Familiarity]	• Need for resource allocation • Fixed allocation (TDM, FDM, WDM) versus dynamic allocation • End-to-end versus network assisted approaches • Fairness • Principles of congestion control • Approaches to Congestion (e.g., Content Distribution Networks)
Readings : [KR13]	

Unit 7: Mobility (5)	
Competences Expected: C1,C7	
Learning Outcomes	**Topics**
• Describe the organization of a wireless network [Familiarity] • Describe how wireless networks support mobile users [Familiarity]	• Principles of cellular networks • 802.11 networks • Issues in supporting mobile nodes (home agents)
Readings : [KR13]	

Unit 8: Social Networking (5)	
Competences Expected: C1,CS2,CS8	
Learning Outcomes	**Topics**
• Discuss the key principles (such as membership, trust) of social networking [Familiarity] • Describe how existing social networks operate [Familiarity] • Construct a social network graph from network data [Usage] • Analyze a social network to determine who the key people are [Usage] • Evaluate a given interpretation of a social network question with associated data [Familiarity]	• Social networks overview • Example social network platforms • Structure of social network graphs • Social network analysis
Readings : [KR13]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS3101. Competitive Programming (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Cor+09] T. H. Cormen et al. *Introduction to Algorithms.* MIT Press, 2009.

6. **Information about the course**

(a) **Brief description about the course** Competitive Programming combines problem-solving challenges with the fun of competing with others. It teaches participants to think faster and develop problem-solving skills that are in high demand in the industry. This course will teach you to solve algorithmic problems quickly by combining theory of algorithms and data structures with practice solving problems.

(b) **Prerrequisites:** CS2102. Analysis and Design of Algorithms. (5^{th} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- That the student uses techniques of data structures and complex algorithms..
- That the student apply the concepts learned for the application on a real problem.
- That the student investigate the possibility of creating a new algorithm and / or new technique to solve a real problem.

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

9. **Competences (IEEE)**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a,b**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a,b**

C24. Understanding the need for lifelong learning and improving skills and abilities.⇒ **Outcome h**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a,b**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a,b**

C24. Understanding the need for lifelong learning and improving skills and abilities.⇒ **Outcome h**

10. List of topics

1. Primera Unidad

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Primera Unidad (20)	
Competences Expected: C24,C1	
Learning Outcomes	**Topics**
• Learning to select the right algorithms for a given problem, integrating multiple algorithms for solving a complex problem. [Usage] • Design new algorithms for real-world problem solving.[Usage]	• Data structure • Dynamic programming • Graph-based algorithms • Sort algorithms
Readings : [Cor+09]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS3102. Advanced Data Structures (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Chá+01] E. Chávez et al. “Proximity Searching in Metric Spaces”. In: *ACM Computing Surveys* 33.3 (Sept. 2001), pp. 273–321.

[Cua+04] Ernesto Cuadros-Vargas et al. “Implementing data structures: An incremental approach”. http://socios.spc.org.pe/ecuadros/cursos/pdfs/. 2004.

[Gam+94] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software.* Computing Series. ISBN-10: 0201633612. Addison-Wesley Professional, Nov. 1994.

[GG98] Volker Gaede and Oliver Günther. “Multidimensional Access Methods”. In: *ACM Computing Surveys* 30.2 (1998), pp. 170–231.

[Knu07a] Donald Ervin Knuth. *The Art of Computer Programming, Fundamental Algorithms.* 3rd. Vol. I. 0-201-89683-4. Addison-Wesley, Feb. 2007.

[Knu07b] Donald Ervin Knuth. *The Art of Computer Programming, Sorting and Searching.* 2nd. Vol. II. 0-201-89685-0. Addison-Wesley, Feb. 2007.

[PI06] Trevor Darrell PGregory Shakhnarovich and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice.* 1st. ISBN 0-262-19547-X. MIT Press, Mar. 2006.

[Sam06] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures.* Illustrated. Elsevier/Morgan Kaufmann, 2006. ISBN: 9780123694461. URL: `http://books.google.com.pe/books?id=vO-NRRKHG84C`.

[Tra+00] C. Traina Jr et al. “Slim-Trees: High Performance Metric Trees Minimizing Overlap between Nodes”. In: *Advances in Database Technology - EDBT 2000, 6th International Conference on Extending Database Technology.* Vol. 1777. Lecture Notes in Computer Science. Konstanz, Germany: Springer, Mar. 2000, pp. 51–65.

[Zez+07] Pavel Zezula et al. *Similarity Search: The Metric Space Approach.* 1st. ISBN-10: 0387291466. Springer, Nov. 2007.

6. **Information about the course**

   (a) **Brief description about the course** Algorithms and data structures are a fundamental part of computer science that allow us to organize information in a more efficient way, so it is important for every professional in the area to have a solid background in this regard.

   In the course of advanced data structures our goal is for the student to know and analyze complex structures, such as Multidimensional Access Methods, Space-Time Access Methods and Metric Access Methods, etc.

   (b) **Prerrequisites:** CS2102. Analysis and Design of Algorithms. (5th Sem)

   (c) **Type of Course:** Mandatory

   (d) **Modality:** Face to face

7. **Specific goals of the Course**

- That the student understands, designs, implements, applies and Propose innovative data structures to solve problems related to the handling of multidimensional data, retrieval of information by similarity, search engines and other computational problems.

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Familiarity**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Familiarity**)

a) An ability to apply knowledge of mathematics, science. (**Familiarity**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Familiarity**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome b**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome c**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome b**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome c**

10. List of topics

1. Técnicas Básicas de Implementación de Estructuras de Datos
2. Métodos de Acceso Multidimensionales
3. Métodos de Acceso Métrico
4. Métodos de Acceso Aproximados
5. Seminarios

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

Unit 1: Técnicas Básicas de Implementación de Estructuras de Datos (16)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• That the student understands the basic differences that involve the different techniques of implementation of data structures[Usage] • That the student analyze the advantages and disadvantages of each of the existing techniques[Usage]	• Structured Programming • Object-oriented programming • Abstract Data Types • Independence of the user programming language of the structure • Platform Independence • Concurrency control • Data Protection • Encapsulation levels (struct, class, namespace, etc)
Readings : [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

Unit 2: Métodos de Acceso Multidimensionales (16)	
Competences Expected: C20	
Learning Outcomes	**Topics**
• That the student understands to know and implement some Access Methods for multidimensional data and temporal space[Usage] • That the student understands the potential of these Access Methods in the future of commercial databases[Usage]	• Access Methods for Point Data • Access Methods for non-point data • Problems with dimension enhancement
Readings : [Sam06], [GG98]	

Unit 3: Métodos de Acceso Métrico (20)	
Competences Expected: C24	
Learning Outcomes	**Topics**
• That the student understands to know and implement some methods of metric access[Usage] • That the student understands the importance of these Access Methods for Information Retrieval by similarity[Usage]	• Metric Access Methods for discrete distances • Metric Access Methods for Continuous Distances
Readings : [Sam06], [Chá+01], [Tra+00], [Zez+07]	

Unit 4: Métodos de Acceso Aproximados (20)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• That the student understands to know and implement some approximate access methods[Usage] • That the student understands the importance of these Access Methods for Information Retrieval by Similarity in environments where Scalability is a very important factor [Usage]	• Space Filling Curves • Locality Sensitive Hashing
Readings : [PI06], [Zez+07], [Sam06]	

Unit 5: Seminarios (8)	
Competences Expected: C20	
Learning Outcomes	**Topics**
• That the student can discuss the latest advances in access methods for different domains of knowledge [Usage]	• Access Methods Temporary Space • Generic Data Structures
Readings : [Sam06], [Chá+01]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS3903. Information systems (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[PM14] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2014.

[Som10] Ian Sommerville. *Software Engineering*. 9th. Addison-Wesley, Mar. 2010.

6. **Information about the course**

(a) **Brief description about the course** Analyze techniques for the correct implementation of scalable, robust, reliable and efficient information systems in organizations.

(b) **Prerrequisites:** CS2901. Software Engineering I. (5^{th} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Implement correctly (scalable, robust, reliable and efficient) Information Systems in organizations.

8. **Contribution to Outcomes**

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

k) Apply the principles of development and design in the construction of software systems of variable complexity. (**Assessment**)

9. **Competences (IEEE)**

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome c**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome c**

C16. Ability to identify advanced computing topics and understanding the frontiers of the discipline.⇒ **Outcome k**

CS4. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.⇒ **Outcome k**

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome i**

CS10. Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems. This should include tools for software control including version control and configuration management.⇒ **Outcome k**

10. List of topics

1. Introduction
2. Strategy
3. Implementation

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Introduction (15)	
Competences Expected: C7,C8	
Learning Outcomes	**Topics**
• Correctly apply technology for information management [Assessment]	• Introduction to information management. • Software for information management. • Technology for information management.
Readings : [Som10], [PM14]	

Unit 2: Strategy (15)	
Competences Expected: C16, CS4	
Learning Outcomes	**Topics**
• Apply and evaluate correctly management strategies [Assessment]	• Strategy for information management. • Strategy for knowledge management • Strategy for information system.
Readings : [Som10], [PM14]	

Unit 3: Implementation (15)	
Competences Expected: CS4, CS6, CS10	
Learning Outcomes	**Topics**
• Implement and correctly evaluate implementation strategies [Assessment]	• Management Information Systems Development. • Change management • Information Architecture
Readings : [Som10], [PM14]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS2H01. Computer Human Interaction (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 1 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Bux07] Bill Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., 2007.

[Dix+04] Alan Dix et al. *Human-computer Interaction*. 3 ed. Prentice-Hall, Inc, 2004.

[Joh10] Jeff Johnson. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. 3 ed. Morgan Kaufmann Publishers Inc., 2010.

[LS06] M. Leavitt and B. Shneiderman. *Research-Based Web Design & Usability Guidelines*. Health and Human Services Dept, 2006.

[Mat11] Lukas Mathis. *Designed for Use: Create Usable Interfaces for Applications and the Web*. Pragmatic Bookshelf, 2011.

[Nor04] Donald A. Norman. *Emotional Design: Why We Love (or Hate) Everyday Things*. Basic Book, 2004.

[RS11] Y. Rogers and J Sharp H. & Preece. *Interaction Design: Beyond Human-Computer Interaction*. 3 ed. John Wiley and Sons Ltd, 2011.

[Sto+05] D. Stone et al. *User Interface Design and Evaluation*. Morgan Kaufmann Series in Interactive Technologies, 2005.

[WW11] D. Wigdor and D. Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann Publishers Inc, 2011.

6. **Information about the course**

   (a) **Brief description about the course** Language has been one of the most significant creations of humanity. From body language and gesture, through verbal and written communication, to iconic symbolic codes and others, it has made possible complex interactions Among humans and facilitated considerably the communication of information. With the invention of automatic and semi-automatic devices, including computers, The need for languages or interfaces to be able to interact with them, has gained great importance. The utility of the software, coupled with user satisfaction and increased productivity, depends on the effectiveness of the User-Computer Interface. So much so, that often the interface is the most important factor in the success and failure of any computer system. The design and implementation of appropriate Human-Computer Interfaces, which in addition to complying with the technical requirements and the transactional logic of the application, consider the subtle psychological implications, sciences and user facilities, It consumes a good part of the life cycle of a software project, and requires specialized skills, both for the construction of the same, and for the performance of usability tests.

   (b) **Prerrequisites:** CS3903. Information systems. (6^{th} Sem)

   (c) **Type of Course:** Mandatory

   (d) **Modality:** Face to face

7. **Specific goals of the Course**

- Know and apply criteria of usability and accessibility to the design and construction of human-computer interfaces, always looking for technology to adapt to people and not people to technology.

- That the student has a vision focused on the user experience by applying appropriate conceptual and technological approaches.

- Understand how emerging technology makes possible new styles of interaction.

- Determine the basic requirements at the interface level, hardware and software for the construction of immersive environments.

8. Contribution to Outcomes

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Familiarity**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Assessment**)

d) An ability to function on multidisciplinary teams. (**Usage**)

o) Improve the conditions of society by putting technology at the service of the human being. (**Familiarity**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Familiarity**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Assessment**)

d) An ability to function on multidisciplinary teams. (**Usage**)

o) Improve the conditions of society by putting technology at the service of the human being. (**Familiarity**)

9. Competences (IEEE)

CS8. Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, multimedia systems and mobile systems..⇒ **Outcome b**

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome c**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome o**

C15. Understanding of the essential concept of process as it relates to professional activity, especially the relationship between product quality and the deployment of appropriate human processes during product development.⇒ **Outcome d**

CS10. Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems. This should include tools for software control including version control and configuration management.⇒ **Outcome d**

CS8. Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, multimedia systems and mobile systems..⇒ **Outcome b**

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome c**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome o**

C15. Understanding of the essential concept of process as it relates to professional activity, especially the relationship between product quality and the deployment of appropriate human processes during product development.⇒ **Outcome d**

CS10. Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems. This should include tools for software control including version control and configuration management.⇒ **Outcome d**

10. List of topics

1. Foundations
2. Factores Humanos
3. User-centered design and testing
4. Designing Interaction
5. New Interactive Technologies
6. Collaboration and communication

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Foundations (8)	
Competences Expected: CS8	
Learning Outcomes	**Topics**
• Discuss why human-centered software development is important [Familiarity] • Define a user-centered design process that explicitly takes account of the fact that the user is not like the developer or their acquaintances [Familiarity] • Summarize the basic precepts of psychological and social interaction [Familiarity] • Develop and use a conceptual vocabulary for analyzing human interaction with software: affordance, conceptual model, feedback, and so forth [Familiarity]	• Contexts for HCI (anything with a user interface, e.g., webpage, business applications, mobile applications, and games) • Usability heuristics and the principles of usability testing • Processes for user-centered development, e.g., early focus on users, empirical testing, iterative design • Principles of good design and good designers; engineering tradeoffs • Different measures for evaluation, e.g., utility, efficiency, learnability, user satisfaction
Readings : [Dix+04], [Sto+05], [RS11]	

Unit 2: Factores Humanos (8)	
Competences Expected: CS8	
Learning Outcomes	**Topics**
• Create and conduct a simple usability test for an existing software application [Familiarity]	• Cognitive models that inform interaction design, e.g., attention, perception and recognition, movement, and memory; gulfs of expectation and execution • Physical capabilities that inform interaction design, e.g., color perception, ergonomics • Accessibility, e.g., interfaces for differently-abled populations (e.g., blind, motion-impaired) • Interfaces for differently-aged population groups (e.g., children, 80+)
Readings : [Dix+04], [Sto+05], [RS11], [Mat11], [Nor04]	

Unit 3: User-centered design and testing (16)	
Competences Expected: C7, CS8, CS10	
Learning Outcomes	**Topics**
• Conduct a quantitative evaluation and discuss/report the results [Familiarity] • For an identified user group, undertake and document an analysis of their needs [Familiarity] • Discuss at least one national or international user interface design standard [Familiarity] • Explain how user-centred design complements other software process models [Familiarity] • Use lo-fi (low fidelity) prototyping techniques to gather, and report, user responses [Usage] • Choose appropriate methods to support the development of a specific UI [Assessment] • Use a variety of techniques to evaluate a given UI [Assessment] • Compare the constraints and benefits of different evaluative methods [Assessment]	• Approaches to, and characteristics of, the design process • Functionality and usability requirements • Techniques for gathering requirements, e.g., interviews, surveys, ethnographic and contextual enquiry • Techniques and tools for the analysis and presentation of requirements, e.g., reports, personas • Task analysis, including qualitative aspects of generating task analytic models • Consideration of HCI as a design discipline – Sketching – Participatory design – Sketching. – Participatory Design. • Prototyping techniques and tools, e.g., sketching, storyboards, low-fidelity prototyping, wireframes • Low-fidelity (paper) prototyping • Quantitative evaluation techniques, e.g., keystroke-level evaluation • Evaluation without users, using both qualitative and quantitative techniques, e.g., walkthroughs, GOMS, expert-based analysis, heuristics, guidelines, and standard • Evaluation with users, e.g., observation, think-aloud, interview, survey, experiment • Challenges to effective evaluation, e.g., sampling, generalization • Reporting the results of evaluations • Internationalization, designing for users from other cultures, cross-cultural
Readings : [Dix+04], [Sto+05], [RS11], [Mat11], [Bux07]	

Unit 4: Designing Interaction (8)	
Competences Expected: CS8, CS15	
Learning Outcomes	**Topics**
• Create a simple application, together with help and documentation, that supports a graphical user interface [Usage]	• Principles of graphical user interfaces (GUIs) • Elements of visual design (layout, color, fonts, labeling) • Handling human/system failure • User interface standards • Presenting information: navigation, representation, manipulation • Interface animation techniques (e.g., scene graphs) • Widget classes and libraries • Internationalization, designing for users from other cultures, cross-cultural • Choosing interaction styles and interaction techniques
Readings : [Dix+04], [Sto+05], [RS11], [Joh10], [Mat11], [LS06]	

Unit 5: New Interactive Technologies (8)	
Competences Expected: C9	
Learning Outcomes	**Topics**
• Describe when non-mouse interfaces are appropriate [Familiarity] • Understand the interaction possibilities beyond mouse-and-pointer interfaces [Familiarity] • Discuss the advantages (and disadvantages) of non-mouse interfaces [Usage] • Describe the optical model realized by a computer graphics system to synthesize stereoscopic view [Familiarity] • Describe the principles of different viewer tracking technologies [Familiarity] • Determine the basic requirements on interface, hardware, and software configurations of a VR system for a specified application [Assessment]	• Choosing interaction styles and interaction techniques • Approaches to design, implementation and evaluation of non-mouse interaction – Touch and multi-touch interfaces – Shared, embodied, and large interfaces – New input modalities (such as sensor and location data) – New Windows, e.g., iPhone, Android – Speech recognition and natural language processing – Wearable and tangible interfaces – Persuasive interaction and emotion – Ubiquitous and context-aware interaction technologies (Ubicomp) – Bayesian inference (e.g. predictive text, guided pointing) – Ambient/peripheral display and interaction • Output – Sound – Stereoscopic display – Force feedback simulation, haptic devices • System architectures – Game engines – Mobile augmented reality – Flight simulators – CAVEs – Medical imaging
Readings : [Dix+04], [Sto+05], [RS11], [WW11], [Mat11]	

Unit 6: Collaboration and communication (8)	
Competences Expected: CS8, CS9	
Learning Outcomes	**Topics**
• Describe the difference between synchronous and asynchronous communication [Familiarity] • Compare the HCI issues in individual interaction with group interaction [Familiarity] • Discuss several issues of social concern raised by collaborative software [Usage] • Discuss the HCI issues in software that embodies human intention [Assessment]	• Asynchronous group communication, e.g., e-mail, forums, social networks • Social media, social computing, and social network analysis • Online collaboration, ’smart’ spaces, and social coordination aspects of workflow technologies • Online communities • Software characters and intelligent agents, virtual worlds and avatars • Social psychology
Readings : [Dix+04], [Sto+05], [RS11]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS341. Programming languages (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 2 HP; 2 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[RH04] Peter Van Roy and Seif Haridi. *Concepts, Techniques, and Models of Computer Programming.* Cambridge, MA, USA: MIT Press, 2004. ISBN: 0262220695.

[Seb12] Robert W. Sebesta. *Concepts of Programming Languages.* 10th. USA: Addison-Wesley Publishing Company, 2012. ISBN: 0131395319.

[Web10] Adam Brooks Webber. *Modern Programming Languages: A Practical Introduction.* 2nd. Franklin, Beedle and Associates, Inc, 2010. ISBN: 978-1-59028-250-2.

6. **Information about the course**

(a) **Brief description about the course** Los lenguajes de programación son el medio a través del cual los programadores describen con precisión los conceptos, formulan algoritmos y representan sus soluciones. Un científico de la computación trabajará con diferentes lenguajes, por separado o en conjunto. Los científicos de la computación deben entender los modelos de programación de los diferentes lenguajes, tomar decisiones de diseño basados en el lenguaje de programación y sus conceptos. El profesional a menudo necesitará aprender nuevos lenguajes y construcciones de programación y debe entender los fundamentos de como las características del lenguaje de programación estan definidas, compuestas e implementadas. El uso eficaz de los lenguajes de programación y la apreciación de sus limitaciones, también requiere un conocimiento básico de traducción de lenguajes de programación y su análisis de ambientes estáticos y dinámicos, así como los componentes de tiempo de ejecución tales como la gestión de memoria, entre otros detalles de relevancia.

(b) **Prerrequisites:** CS2101. Theory of Computation. (4^{th} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Capacitar a los estudiantes para entender los lenguajes de programación desde diferentes tipos de vista, según el modelo subyacente, los componentes fundamentales presentes en todo lenguaje de programación y como objetos formales dotados de una estructura y un significado según diversos enfoques.

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

9. Competences (IEEE)

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a**

CS1. Model and design computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.⇒ **Outcome b**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome i**

CS3. Analyze the extent to which a computer-based system meets the criteria defined for its current use and future development.⇒ **Outcome j**

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome j**

10. List of topics

1.
2. Language Pragmatics
3. Type Systems
4. Object-Oriented Programming
5. Functional Programming
6. Event-Driven and Reactive Programming
7. Logic Programming

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: (18)	
Competences Expected: C2, CS1, CS2	
Learning Outcomes	**Topics**
• Reconocer el desarrollo histórico de los lenguajes de programación. [Familiarity] • Identificar los paradigmas que agrupan a la mayoría de lenguajes de programación existentes hoy en día. [Familiarity] • Explain how programs that process other programs treat the other programs as their input data [Familiarity] • Describe an abstract syntax tree for a small language [Familiarity] • Write a program to process some representation of code for some purpose, such as an interpreter, an expression optimizer, or a documentation generator [Usage] • Distinguish a language definition (what constructs mean) from a particular language implementation (compiler vs interpreter, run-time representation of data objects, etc) [Familiarity] • Reconocer como funciona un programa a nivel de computador. [Familiarity]	• Historia de los Lenguajes de Programación • Programs that take (other) programs as input such as interpreters, compilers, type-checkers, documentation generators • Data structures to represent code for execution, translation, or transmission • Estructura de un programa: Léxico, Sintáctico y Semántico • BNF • Interpretation vs. compilation to native code vs. compilation to portable intermediate representation [Familiarity]
Readings : [Seb12], [Web10]	

Unit 2: Language Pragmatics (12)	
Competences Expected: C2, CS1, CS2	
Learning Outcomes	**Topics**
• Discuss the role of concepts such as orthogonality and well-chosen defaults in language design [Usage] • Use crisp and objective criteria for evaluating language-design decisions [Usage] • Give an example program whose result can differ under different rules for evaluation order, precedence, or associativity [Usage] • Show uses of delayed evaluation, such as user-defined control abstractions [Familiarity] • Discuss the need for allowing calls to external calls and system libraries and the consequences for language implementation [Familiarity]	• Principles of language design such as orthogonality • Evaluation order, precedence and associativity • Eager vs. delayed evaluation • Defining control and iteration constructs • External calls and system libraries
Readings : [Seb12], [Web10], [RH04]	

Unit 3: Type Systems (18)	
Competences Expected: C2, CS1, CS2	
Learning Outcomes	**Topics**
• Define a type system precisely and compositionally [Usage] • For various foundational type constructors, identify the values they describe and the invariants they enforce [Familiarity] • Precisely specify the invariants preserved by a sound type system [Familiarity] • Prove type safety for a simple language in terms of preservation and progress theorems [Usage] • Implement a unification-based type-inference algorithm for a simple language [Usage] • Explain how static overloading and associated resolution algorithms influence the dynamic behavior of programs [Familiarity]	• Compositional type constructors, such as product types (for aggregates), sum types (for unions), function types, quantified types, and recursive types • Type checking • Type safety as preservation plus progress • Type inference • Static overloading
Readings : [Seb12], [Web10], [RH04]	

Unit 4: Object-Oriented Programming (12)	
Competences Expected: CS2, CS3, CS6	
Learning Outcomes	**Topics**
• Design and implement a class [Usage] • Use subclassing to design simple class hierarchies that allow code to be reused for distinct subclasses [Usage] • Correctly reason about control flow in a program using dynamic dispatch [Usage] • Compare and contrast (1) the procedural/functional approach—defining a function for each operation with the function body providing a case for each data variant—and (2) the object-oriented approach—defining a class for each data variant with the class definition providing a method for each operation Understand both as defining a matrix of operations and variants [Assessment] • Explain the relationship between object-oriented inheritance (code-sharing and overriding) and subtyping (the idea of a subtype being usable in a context that expects the supertype) [Usage] • Use object-oriented encapsulation mechanisms such as interfaces and private members [Usage] • Define and use iterators and other operations on aggregates, including operations that take functions as arguments, in multiple programming languages, selecting the most natural idioms for each language [Usage]	• Object-oriented design – Decomposition into objects carrying state and having behavior – Class-hierarchy design for modeling • Definition of classes: fields, methods, and constructors • Subclasses, inheritance, and method overriding • Dynamic dispatch: definition of method-call • Subtyping – Subtype polymorphism; implicit upcasts in typed languages – Notion of behavioral replacement: subtypes acting like supertypes – Relationship between subtyping and inheritance • Object-oriented idioms for encapsulation – Privacy and visibility of class members – Interfaces revealing only method signatures – Abstract base classes • Using collection classes, iterators, and other common library components
Readings : [Seb12], [Web10], [RH04]	

Unit 5: Functional Programming (18)	
Competences Expected: CS2, CS3, CS6	
Learning Outcomes	**Topics**
• Write basic algorithms that avoid assigning to mutable state or considering reference equality [Usage] • Write useful functions that take and return other functions [Usage] • Compare and contrast (1) the procedural/functional approach-defining a function for each operation with the function body providing a case for each data variant-and (2) the object-oriented approach-defining a class for each data variant with the class definition providing a method for each operation Understand both as defining a matrix of operations and variants [Assessment] • Correctly reason about variables and lexical scope in a program using function closures [Usage] • Use functional encapsulation mechanisms such as closures and modular interfaces [Usage] • Define and use iterators and other operations on aggregates, including operations that take functions as arguments, in multiple programming languages, selecting the most natural idioms for each language [Usage]	• Effect-free programming – Function calls have no side effects, facilitating compositional reasoning – Variables are immutable, preventing unexpected changes to program data by other code – Data can be freely aliased or copied without introducing unintended effects from mutation • Processing structured data (e.g., trees) via functions with cases for each data variant – Associated language constructs such as discriminated unions and pattern-matching over them – Functions defined over compound data in terms of functions applied to the constituent pieces • First-class functions (taking, returning, and storing functions) • Function closures (functions using variables in the enclosing lexical environment) – Basic meaning and definition – creating closures at run-time by capturing the environment – Canonical idioms: call-backs, arguments to iterators, reusable code via function arguments – Using a closure to encapsulate data in its environment – Currying and partial application • Defining higher-order operations on aggregates, especially map, reduce/fold, and filter
Readings : [Seb12], [Web10], [RH04]	

Unit 6: Event-Driven and Reactive Programming (12)	
Competences Expected: CS2, CS3, CS6	
Learning Outcomes	**Topics**
• Write event handlers for use in reactive systems, such as GUIs [Usage] • Explain why an event-driven programming style is natural in domains where programs react to external events [Familiarity] • Describe an interactive system in terms of a model, a view, and a controller [Familiarity]	• Events and event handlers • Canonical uses such as GUIs, mobile devices, robots, servers • Using a reactive framework – Defining event handlers/listeners – Main event loop not under event-handler-writer's control • Externally-generated events and program-generated events • Separation of model, view, and controller
Readings : [Seb12]	

Unit 7: Logic Programming (12)	
Competences Expected: CS2, CS3, CS6	
Learning Outcomes	**Topics**
• Use a logic language to implement a conventional algorithm [Usage] • Use a logic language to implement an algorithm employing implicit search using clauses, relations, and cuts [Usage]	• Causal representation of data structures and algorithms • Unification • Bactracking and search • Cuts
Readings : [Seb12], [Web10], [RH04]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: CS3P01. Parallel and Distributed Computing (Mandatory)
2. Credits: 4
3. Hours of theory and Lab: 2 HT; 4 HL; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[KH13] David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach.* 2nd. Morgan Kaufmann, 2013. ISBN: 978-0-12-415992-1.

[Mat14] Norm Matloff. *Programming on Parallel Machines.* University of California, Davis, 2014. URL: `http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf`.

[Pac11] Peter S. Pacheco. *An Introduction to Parallel Programming.* 1st. Morgan Kaufmann, 2011. ISBN: 978-0-12-374260-5.

[Qui03] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP.* 1st. McGraw-Hill Education Group, 2003. ISBN: 0071232656.

[SK10] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming.* 1st. Addison-Wesley Professional, 2010. ISBN: 0131387685, 9780131387683.

6. Information about the course

(a) **Brief description about the course** The last decade has brought explosive growth in computing with multiprocessors, including Multi-core processors and distributed data centers. As a result, computing parallel and distributed has become a widely elective subject to be one of the main components in the mesh studies in computer science undergraduate. Both parallel and distributed computing the simultaneous execution of multiple processes, whose operations have the potential to intercalar in a complex way. Parallel and distributed computing builds on foundations in many areas, including understanding the fundamental concepts of systems, such as: concurrency and parallel execution, consistency in state / memory manipulation, and latency. The communication and coordination between processes has its foundations in the passage of messages and models of shared memory of computing and algorithmic concepts like atomicity, consensus and conditional waiting.Achieving acceleration in practice requires an understanding of parallel algorithms, strategies for decomposition problem, systems architecture, implementation strategies and analysis of performance. Distributed systems highlight the problems of security and tolerance to Failures, emphasize the maintenance of the replicated state and introduce additional problems in the field of computer networks.

(b) **Prerrequisites:**

- CS2102. Analysis and Design of Algorithms. (5^{th} Sem)
- CS2301. Networking and Communication. (6^{th} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- That the student is able to create parallel applications of medium complexity by efficiently leveraging machines with multiple cores.

- That the student is able to compare sequential and parallel applications.
- That the student is able to convert, when the situation warrants, sequential applications to parallel efficiently

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

9. Competences (IEEE)

C3. An intellectual understanding of, and an appreciation for, the central role of algorithms and data structures.⇒ **Outcome b,g**

C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.⇒ **Outcome b,c**

CS1. Model and design computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.⇒ **Outcome a**

CS4. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.⇒ **Outcome a**

C3. An intellectual understanding of, and an appreciation for, the central role of algorithms and data structures.⇒ **Outcome b,g**

C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.⇒ **Outcome b,c**

CS1. Model and design computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.⇒ **Outcome a**

CS4. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.⇒ **Outcome a**

10. List of topics

1. Parallelism Fundamentals
2. Parallel Architecture
3. Parallel Decomposition
4. Communication and Coordination
5. Parallel Algorithms, Analysis, and Programming
6. Parallel Performance

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Parallelism Fundamentals (18)	
Competences Expected: C2	
Learning Outcomes	**Topics**
• Distinguish using computational resources for a faster answer from managing efficient access to a shared resource [Familiarity] • Distinguish multiple sufficient programming constructs for synchronization that may be inter-implementable but have complementary advantages [Familiarity] • Distinguish data races from higher level races [Familiarity]	• Multiple simultaneous computations • Goals of parallelism (e.g., throughput) versus concurrency (e.g., controlling access to shared resources) • Parallelism, communication, and coordination – Parallelism, communication, and coordination – Need for synchronization • Programming errors not found in sequential programming – Data races (simultaneous read/write or write/write of shared state) – Higher-level races (interleavings violating program intention, undesired non-determinism) – Lack of liveness/progress (deadlock, starvation)
Readings : [Pac11], [Mat14], [Qui03]	

Unit 2: Parallel Architecture (12)	
Competences Expected: C4	
Learning Outcomes	**Topics**
• Explain the differences between shared and distributed memory [Assessment] • Describe the SMP architecture and note its key features [Assessment] • Characterize the kinds of tasks that are a natural match for SIMD machines [Usage] • Describe the advantages and limitations of GPUs vs CPUs [Usage] • Explain the features of each classification in Flynn's taxonomy [Usage] • Describe the challenges in maintaining cache coherence [Familiarity] • Describe the key performance challenges in different memory and distributed system topologies [Familiarity]	• Multicore processors • Shared vs distributed memory • Symmetric multiprocessing (SMP) • SIMD, vector processing • GPU, co-processing • Flynn's taxonomy • Instruction level support for parallel programming – Atomic instructions such as Compare and Set • Memory issues – Multiprocessor caches and cache coherence – Non-uniform memory access (NUMA) • Topologies – Interconnects – Clusters – Resource sharing (e.g., buses and interconnects)
Readings : [Pac11], [KH13], [SK10]	

Unit 3: Parallel Decomposition (18)	
Competences Expected: C16	
Learning Outcomes	**Topics**
• Explain why synchronization is necessary in a specific parallel program [Usage] • Identify opportunities to partition a serial program into independent parallel modules [Familiarity] • Write a correct and scalable parallel algorithm [Usage] • Parallelize an algorithm by applying task-based decomposition [Usage] • Parallelize an algorithm by applying data-parallel decomposition [Usage] • Write a program using actors and/or reactive processes [Usage]	• Need for communication and coordination/synchronization • Independence and partitioning • Basic knowledge of parallel decomposition concept • Task-based decomposition – Implementation strategies such as threads • Data-parallel decomposition – Strategies such as SIMD and MapReduce • Actors and reactive processes (e.g., request handlers)
Readings : [Pac11], [Mat14], [Qui03]	

<table>
<tr><td colspan="2"><b>Unit 4: Communication and Coordination (18)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: C16</b></td></tr>
<tr><td><b>Learning Outcomes</b></td><td><b>Topics</b></td></tr>
<tr><td>• Use mutual exclusion to avoid a given race condition [Usage]
• Give an example of an ordering of accesses among concurrent activities (eg, program with a data race) that is not sequentially consistent [Familiarity]
• Give an example of a scenario in which blocking message sends can deadlock [Usage]
• Explain when and why multicast or event-based messaging can be preferable to alternatives [Familiarity]
• Write a program that correctly terminates when all of a set of concurrent tasks have completed [Usage]
• Give an example of a scenario in which an attempted optimistic update may never complete [Familiarity]
• Use semaphores or condition variables to block threads until a necessary precondition holds [Usage]</td><td>• Shared Memory
• Consistency, and its role in programming language guarantees for data-race-free programs
• Message passing
– Point-to-point versus multicast (or event-based) messages
– Blocking versus non-blocking styles for sending and receiving messages
– Message buffering (cross-reference PF/Fundamental Data Structures/Queues)
• Atomicity
– Specifying and testing atomicity and safety requirements
– Granularity of atomic accesses and updates, and the use of constructs such as critical sections or transactions to describe them
– Mutual Exclusion using locks, semaphores, monitors, or related constructs
∗ Potential for liveness failures and deadlock (causes, conditions, prevention)
– Composition
∗ Composing larger granularity atomic actions using synchronization
∗ Transactions, including optimistic and conservative approaches
• Consensus
– (Cyclic) barriers, counters, or related constructs
• Conditional actions
– Conditional waiting (e.g., using condition variables)</td></tr>
<tr><td colspan="2"><b>Readings :</b> [Pac11], [Mat14], [Qui03]</td></tr>
</table>

Unit 5: Parallel Algorithms, Analysis, and Programming (18)	
Competences Expected: CS2	
Learning Outcomes	**Topics**
• Define “critical path”, “work”, and “span” [Familiarity] • Compute the work and span, and determine the critical path with respect to a parallel execution diagram [Usage] • Define “speed-up” and explain the notion of an algorithm’s scalability in this regard [Familiarity] • Identify independent tasks in a program that may be parallelized [Usage] • Characterize features of a workload that allow or prevent it from being naturally parallelized [Familiarity] • Implement a parallel divide-and-conquer (and/or graph algorithm) and empirically measure its performance relative to its sequential analog [Usage] • Decompose a problem (eg, counting the number of occurrences of some word in a document) via map and reduce operations [Usage] • Provide an example of a problem that fits the producer-consumer paradigm [Usage] • Give examples of problems where pipelining would be an effective means of parallelization [Usage] • Implement a parallel matrix algorithm [Usage] • Identify issues that arise in producer-consumer algorithms and mechanisms that may be used for addressing them [Usage]	• Critical paths, work and span, and the relation to Amdahl’s law • Speed-up and scalability • Naturally (embarrassingly) parallel algorithms • Parallel algorithmic patterns (divide-and-conquer, map and reduce, master-workers, others) – Specific algorithms (e.g., parallel MergeSort) • Parallel graph algorithms (e.g., parallel shortest path, parallel spanning tree) (cross-reference AL/Algorithmic Strategies/Divide-and-conquer) • Parallel matrix computations • Producer-consumer and pipelined algorithms • Examples of non-scalable parallel algorithms
Readings : [Mat14], [Qui03]	

Unit 6: Parallel Performance (18)	
Competences Expected: CS3	
Learning Outcomes	**Topics**
• Detect and correct a load imbalance [Usage] • Calculate the implications of Amdahl's law for a particular parallel algorithm (cross-reference SF/Evaluation for Amdahl's Law) [Usage] • Describe how data distribution/layout can affect an algorithm's communication costs [Familiarity] • Detect and correct an instance of false sharing [Usage] • Explain the impact of scheduling on parallel performance [Familiarity] • Explain performance impacts of data locality [Familiarity] • Explain the impact and trade-off related to power usage on parallel performance [Familiarity]	• Load balancing • Performance measurement • Scheduling and contention (cross-reference OS/Scheduling and Dispatch) • Evaluating communication overhead • Data management – Non-uniform communication costs due to proximity (cross-reference SF/Proximity) – Cache effects (e.g., false sharing) – Maintaining spatial locality • Power usage and management
Readings : [Pac11], [Mat14], [KH13], [SK10]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: CS2501. Computer graphics (Elective)
2. Credits: 4
3. Hours of theory and Lab: 2 HT; 4 HL; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[HB90] Donald Hearn and Pauline Baker. *Computer Graphics in C*. Prentice Hall, 1990.

[Hug+13] John F. Hughes et al. *Computer Graphics - Principles and Practice 3rd Edition*. Addison-Wesley, 2013.

[Shr+13] Dave Shreiner et al. *OpenGL, Programming Guide, Eighth Edition*. Addison-Wesley, 2013.

[Wol11] David Wolff. *OpenGL 4.0 Shading Language Cookbook*. Packt Publishing, 2011.

6. Information about the course

(a) **Brief description about the course** It offers an introduction to the area of Computer Graphics, which is an important part of Computer Science. The purpose of this course is to investigate the fundamental principles, techniques and tools for this area.

(b) **Prerrequisites:** CS3102. Advanced Data Structures . (6^{th} Sem)

(c) **Type of Course:** Elective

(d) **Modality:** Face to face

7. Specific goals of the Course

- Bring students to concepts and techniques used in complex 3-D graphics applications.
- Give the student the necessary tools to determine which graphics software and which platform are best suited to develop a specific application.

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome b**

C5. Ability to implement algorithms and data structures in software.⇒ **Outcome b**

C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.⇒ **Outcome i**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome i**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome b**

C5. Ability to implement algorithms and data structures in software.⇒ **Outcome b**

C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.⇒ **Outcome i**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome i**

10. List of topics

1. Fundamental Concepts
2. Basic Rendering
3. Programming Interactive Systems
4. Geometric Modeling
5. Advanced Rendering
6. Computer Animation

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

<table>
<tr><td colspan="2"><b>Unit 1: Fundamental Concepts (6)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: C1,C2</b></td></tr>
<tr><td><b>Learning Outcomes</b></td><td><b>Topics</b></td></tr>
<tr><td>• Explain in general terms how analog signals can be reasonably represented by discrete samples, for example, how images can be represented by pixels [Familiarity]
• Describe color models and their use in graphics display devices [Familiarity]
• Describe the tradeoffs between storing information vs storing enough information to reproduce the information, as in the difference between vector and raster rendering [Familiarity]
• Describe the basic process of producing continuous motion from a sequence of discrete frames (sometimes called “flicker fusion”) [Familiarity]</td><td>• Media applications including user interfaces, audio and video editing, game engines, cad, visualization, virtual reality
• Tradeoffs between storing data and re-computing data as embodied by vector and raster representations of images
• Additive and subtractive color models (CMYK and RGB) and why these provide a range of colors
• Animation as a sequence of still images</td></tr>
<tr><td colspan="2"><b>Readings :</b> [HB90]</td></tr>
</table>

Unit 2: Basic Rendering (12)	
Competences Expected: C1,C4	
Learning Outcomes	**Topics**
• Discuss the light transport problem and its relation to numerical integration ie, light is emitted, scatters around the scene, and is measured by the eye [Familiarity] • Describe the basic graphics pipeline and how forward and backward rendering factor in this [Familiarity] • Create a program to display 3D models of simple graphics images [Usage] • Obtain 2-dimensional and 3-dimensional points by applying affine transformations [Usage] • Apply 3-dimensional coordinate system and the changes required to extend 2D transformation operations to handle transformations in 3D [Usage] • Contrast forward and backward rendering [Assessment] • Explain the concept and applications of texture mapping, sampling, and anti-aliasing [Familiarity] • Explain the ray tracing/rasterization duality for the visibility problem [Familiarity] • Implement a simple real-time renderer using a rasterization API (eg, OpenGL) using vertex buffers and shaders [Usage] • Compute space requirements based on resolution and color coding [Assessment] • Compute time requirements based on refresh rates, rasterization techniques [Assessment]	• Rendering in nature, e.g., the emission and scattering of light and its relation to numerical integration • Forward and backward rendering (i.e., ray-casting and rasterization) • Basic radiometry, similar triangles, and projection model • Affine and coordinate system transformations • Ray tracing • Visibility and occlusion, including solutions to this problem such as depth buffering, Painter's algorithm, and ray tracing • Simple triangle rasterization • Rendering with a shader-based API • Application of spatial data structures to rendering • Sampling and anti-aliasing • Forward and backward rendering (i.e., ray-casting and rasterization)
Readings : [HB90], [Hug+13], [Wol11], [Shr+13]	

Unit 3: Programming Interactive Systems (2)	
Competences Expected: C8	
Learning Outcomes	**Topics**
• Discuss the advantages (and disadvantages) of non-mouse interfaces [Assessment]	• Event management and user interaction • Approaches to design, implementation and evaluation of non-mouse interaction – Touch and multi-touch interfaces – Shared, embodied, and large interfaces – New input modalities (such as sensor and location data) – New Windows, e.g., iPhone, Android – Speech recognition and natural language processing – Wearable and tangible interfaces – Persuasive interaction and emotion – Ubiquitous and context-aware interaction technologies (Ubicomp) – Bayesian inference (e.g. predictive text, guided pointing) – Ambient/peripheral display and interaction
Readings : [HB90]	

Unit 4: Geometric Modeling (15)	
Competences Expected: C1,C5	
Learning Outcomes	**Topics**
• Represent curves and surfaces using both implicit and parametric forms [Usage] • Create simple polyhedral models by surface tessellation [Usage] • Generate a mesh representation from an implicit surface [Usage] • Generate a mesh from data points acquired with a laser scanner [Usage] • Construct CSG models from simple primitives, such as cubes and quadric surfaces [Usage] • Contrast modeling approaches with respect to space and time complexity and quality of image [Assessment]	• Basic geometric operations such as intersection calculation and proximity tests • Volumes, voxels, and point-based representations • Parametric polynomial curves and surfaces • Implicit representation of curves and surfaces • Approximation techniques such as polynomial curves, Bezier curves, spline curves and surfaces, and nonuniform rational basis (NURB) spines, and level set method • Surface representation techniques including tessellation, mesh representation, mesh fairing, and mesh generation techniques such as Delaunay triangulation, marching cubes • Spatial subdivision techniques • Procedural models such as fractals, generative modeling, and L-systems • Elastically deformable and freeform deformable models • Subdivision surfaces • Multiresolution modeling • Reconstruction • Constructive Solid Geometry (CSG) representation
Readings : [HB90], [Shr+13]	

Unit 5: Advanced Rendering (6)	
Competences Expected: C1,C4	
Learning Outcomes	**Topics**
• Demonstrate how an algorithm estimates a solution to the rendering equation [Assessment] • Prove the properties of a rendering algorithm, eg, complete, consistent, and unbiased [Assessment] • Implement a non-trivial shading algorithm (eg, toon shading, cascaded shadow maps) under a rasterization API [Usage] • Discuss how a particular artistic technique might be implemented in a renderer [Familiarity] • Explain how to recognize the graphics techniques used to create a particular image [Familiarity]	• Time (motion blur), lens position (focus), and continuous frequency (color) and their impact on rendering • Shadow mapping • Occlusion culling • Subsurface scattering • Non-photorealistic rendering • GPU architecture • Human visual systems including adaptation to light, sensitivity to noise, and flicker fusion
Readings : [HB90], [Hug+13], [Wol11], [Shr+13]	

Unit 6: Computer Animation (4)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Compute the location and orientation of model parts using an forward kinematic approach [Usage] • Implement the spline interpolation method for producing in-between positions and orientations [Usage] • Implement algorithms for physical modeling of particle dynamics using simple Newtonian mechanics, for example Witkin & Kass, snakes and worms, symplectic Euler, Stormer/Verlet, or midpoint Euler methods [Usage] • Discuss the basic ideas behind some methods for fluid dynamics for modeling ballistic trajectories, for example for splashes, dust, fire, or smoke [Familiarity] • Use common animation software to construct simple organic forms using metaball and skeleton [Usage]	• Forward and inverse kinematics • Collision detection and response • Procedural animation using noise, rules (boids/crowds), and particle systems • Skinning algorithms • Physics based motions including rigid body dynamics, physical particle systems, mass-spring networks for cloth and flesh and hair • Key-frame animation • Splines • Data structures for rotations, such as quaternions • Camera animation • Motion capture
Readings : [HB90], [Shr+13]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS2601. Artificial intelligence (Elective)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. Bibliography

[De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications.* CRC Press, 2006.

[Gol89] David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley, 1989.

[Hay99] Simon Haykin. *Neural networks: A Comprehensive Foundation.* Prentice Hall, 1999.

[Nil01] Nils Nilsson. *Inteligencia Artificial: Una nueva visión.* McGraw-Hill, 2001.

[Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial.* Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.

[RN03] Stuart Russell and Peter Norvig. *Inteligencia Artifical: Un enfoque moderno.* Prentice Hall, 2003.

6. Information about the course

(a) **Brief description about the course** Research in Artificial Intelligence has led to the development of numerous relevant tonic, aimed at the automation of human intelligence, giving a panoramic view of different algorithms that simulate the different aspects of the behavior and the intelligence of the human being.

(b) **Prerrequisites:** IN0054. Statistics and Probabilities . (4^{th} Sem)

(c) **Type of Course:** Elective

(d) **Modality:** Face to face

7. Specific goals of the Course

- Evaluate the possibilities of simulation of intelligence, for which the techniques of knowledge modeling will be studied.
- Build a notion of intelligence that later supports the tasks of your simulation.

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Familiarity**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Familiarity**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome a**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome j**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome a**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome j**

10. List of topics

1. Fundamental Issues
2. Basic Search Strategies
3. Basic Knowledge Representation and Reasoning
4. Advanced Search
5. Advanced Representation and Reasoning
6. Agents
7. Natural Language Processing
8. Basic Machine Learning
9. Robotics
10. Perception and Computer Vision

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Fundamental Issues (2)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Describe Turing test and the “Chinese Room” thought experiment [Usage] • Determing the characteristics of a given problem that an intelligent systems must solve [Usage]	• Overview of AI problems, examples of successful recent AI applications • What is intelligent behavior? – The Turing test – Rational versus non-rational reasoning • Problem characteristics – Fully versus partially observable – Single versus multi-agent – Deterministic versus stochastic – Static versus dynamic – Discrete versus continuous • Nature of agents – Autonomous versus semi-autonomous – Reflexive, goal-based, and utility-based – The importance of perception and environmental interactions • Philosophical and ethical issues.
Readings : [De 06], [Pon+14]	

Unit 2: Basic Search Strategies (4)	
Competences Expected: C20	
Learning Outcomes	**Topics**
• Formulate an efficient problem space for a problem expressed in natural language (eg, English) in terms of initial and goal states, and operators [Usage] • Describe the role of heuristics and describe the trade-offs among completeness, optimality, time complexity, and space complexity [Usage] • Describe the problem of combinatorial explosion of search space and its consequences [Usage] • Select and implement an appropriate uninformed search algorithm for a problem, and characterize its time and space complexities [Usage] • Select and implement an appropriate informed search algorithm for a problem by designing the necessary heuristic evaluation function [Usage] • Evaluate whether a heuristic for a given problem is admissible/can guarantee optimal solution [Usage] • Formulate a problem specified in natural language (eg, English) as a constraint satisfaction problem and implement it using a chronological backtracking algorithm or stochastic local search [Usage] • Compare and contrast basic search issues with game playing issues [Usage]	• Problem spaces (states, goals and operators), problem solving by search • Factored representation (factoring state into variables) • Uninformed search (breadth-first, depth-first, depth-first with iterative deepening) • Heuristics and informed search (hill-climbing, generic best-first, A*) • Space and time efficiency of search • Two-player games (introduction to minimax search) • Constraint satisfaction (backtracking and local search methods)
Readings : [Nil01], [Pon+14]	

Unit 3: Basic Knowledge Representation and Reasoning (6)	
Competences Expected: C24	
Learning Outcomes	**Topics**
• Translate a natural language (eg, English) sentence into predicate logic statement [Usage] • Convert a logic statement into clause form [Usage] • Apply resolution to a set of logic statements to answer a query [Usage] • Make a probabilistic inference in a real-world problem using Bayes' theorem to determine the probability of a hypothesis given evidence [Usage]	• Review of propositional and predicate logic • Resolution and theorem proving (propositional logic only) • Forward chaining, backward chaining • Review of probabilistic reasoning, Bayes theorem
Readings : [Nil01], [RN03], [Pon+14]	

Unit 4: Advanced Search (4)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Design and implement a genetic algorithm solution to a problem [Usage] • Design and implement a simulated annealing schedule to avoid local minima in a problem [Usage] • Design and implement A*,beam search to solve a problem [Usage] • Apply minimax search with alpha-beta pruning to prune search space in a two-player game [Usage] • Compare and contrast genetic algorithms with classic search techniques [Usage] • Compare and contrast various heuristic searches vis-a-vis applicability to a given problem [Usage]	• Constructing search trees, dynamic search space, combinatorial explosion of search space • Stochastic search – Simulated annealing – Genetic algorithms – Monte-Carlo tree search • Implementation of A* search, beam search • Minimax search, alpha-beta pruning • Expectimax search (MDP-solving) and chance nodes
Readings : [Gol89], [Nil01], [RN03], [Pon+14]	

Unit 5: Advanced Representation and Reasoning (6)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Compare and contrast the most common models used for structured knowledge representation, highlighting their strengths and weaknesses [Usage] • Identify the components of non-monotonic reasoning and its usefulness as a representational mechanisms for belief systems [Usage] • Compare and contrast the basic techniques for representing uncertainty [Usage] • Compare and contrast the basic techniques for qualitative representation [Usage] • Apply situation and event calculus to problems of action and change [Usage] • Explain the distinction between temporal and spatial reasoning, and how they interrelate [Usage] • Explain the difference between rule-based, case-based and model-based reasoning techniques [Usage] • Define the concept of a planning system and how it differs from classical search techniques [Usage]	• Knowledge representation issues – Description logics – Ontology engineering • Non-monotonic reasoning (e.g., non-classical logics, default reasoning) • Argumentation • Reasoning about action and change (e.g., situation and event calculus) • Temporal and spatial reasoning • Rule-based Expert Systems • Semantic networks • Model-based and Case-based reasoning
Readings : [Nil01], [RN03], [Pon+14]	

Unit 6: Agents (6)

Competences Expected: C1

Learning Outcomes

- List the defining characteristics of an intelligent agent [Usage]
- Characterize and contrast the standard agent architectures [Usage]
- Describe the applications of agent theory to domains such as software agents, personal assistants, and believable agents [Usage]
- Describe the primary paradigms used by learning agents [Usage]
- Demonstrate using appropriate examples how multi-agent systems support agent interaction [Usage]

Topics

- Definitions of agents
- Agent architectures (e.g., reactive, layered, cognitive)
- Agent theory
- Rationality, game theory
  - Decision-theoretic agents
  - Markov decision processes (MDP)
- Software agents, personal assistants, and information access
  - Collaborative agents
  - Information-gathering agents
  - Believable agents (synthetic characters, modeling emotions in agents)
- Learning agents
- Multi-agent systems
  - Collaborating agents
  - Agent teams
  - Competitive agents (e.g., auctions, voting)
  - Swarm systems and biologically inspired models

Readings : [Nil01], [RN03], [Pon+14]

Unit 7: Natural Language Processing (4)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Define and contrast deterministic and stochastic grammars, providing examples to show the adequacy of each [Usage] • Simulate, apply, or implement classic and stochastic algorithms for parsing natural language [Usage] • Identify the challenges of representing meaning [Usage] • List the advantages of using standard corpora Identify examples of current corpora for a variety of NLP tasks [Usage] • Identify techniques for information retrieval, language translation, and text classification [Usage]	• Deterministic and stochastic grammars • Parsing algorithms – CFGs and chart parsers (e.g. CYK) – Probabilistic CFGs and weighted CYK • Representing meaning / Semantics – Logic-based knowledge representations – Semantic roles – Temporal representations – Beliefs, desires, and intentions • Corpus-based methods • N-grams and HMMs • Smoothing and backoff • Examples of use: POS tagging and morphology • Information retrieval – Vector space model ∗ TF & IDF – Precision and recall • Information extraction • Language translation • Text classification, categorization – Bag of words model
Readings : [Nil01], [RN03], [Pon+14]	

Unit 8: Basic Machine Learning (10)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• List the differences among the three main styles of learning: supervised, reinforcement, and unsupervised [Usage] • Identify examples of classification tasks, including the available input features and output to be predicted [Usage] • Explain the difference between inductive and deductive learning [Usage] • Describe over-fitting in the context of a problem [Usage] • Apply the simple statistical learning algorithm such as Naive Bayesian Classifier to a classification task and measure the classifier's accuracy [Usage]	• Definition and examples of broad variety of machine learning tasks, including classification • Inductive learning • Simple statistical-based learning, such as Naive Bayesian Classifier, decision trees • The over-fitting problem • Measuring classifier accuracy
Readings : [Hay99], [Nil01], [RN03], [Pon+14]	

Unit 9: Robotics (6)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• List capabilities and limitations of today's state-of-the-art robot systems, including their sensors and the crucial sensor processing that informs those systems [Usage] • Integrate sensors, actuators, and software into a robot designed to undertake some task [Usage] • Program a robot to accomplish simple tasks using deliberative, reactive, and/or hybrid control architectures [Usage] • Implement fundamental motion planning algorithms within a robot configuration space [Usage] • Characterize the uncertainties associated with common robot sensors and actuators; articulate strategies for mitigating these uncertainties [Usage] • List the differences among robots' representations of their external environment, including their strengths and shortcomings [Usage] • Compare and contrast at least three strategies for robot navigation within known and/or unknown environments, including their strengths and shortcomings [Usage] • Describe at least one approach for coordinating the actions and sensing of several robots to accomplish a single task [Usage]	• Overview: problems and progress – State-of-the-art robot systems, including their sensors and an overview of their sensor processing – Robot control architectures, e.g., deliberative vs. reactive control and Braitenberg vehicles – World modeling and world models – Inherent uncertainty in sensing and in control • Configuration space and environmental maps • Interpreting uncertain sensor data • Localizing and mapping • Navigation and control • Motion planning • Multiple-robot coordination
Readings : [Nil01], [RN03], [Pon+14]	

Unit 10: Perception and Computer Vision (6)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Summarize the importance of image and object recognition in AI and indicate several significant applications of this technology [Usage] • List at least three image-segmentation approaches, such as thresholding, edge-based and region-based algorithms, along with their defining characteristics, strengths, and weaknesses [Usage] • Implement 2d object recognition based on contour- and/or region-based shape representations [Usage] • Distinguish the goals of sound-recognition, speech-recognition, and speaker-recognition and identify how the raw audio signal will be handled differently in each of these cases [Usage] • Provide at least two examples of a transformation of a data source from one sensory domain to another, eg, tactile data interpreted as single-band 2d images [Usage] • Implement a feature-extraction algorithm on real data, eg, an edge or corner detector for images or vectors of Fourier coefficients describing a short slice of audio signal [Usage] • Implement an algorithm combining features into higher-level percepts, eg, a contour or polygon from visual primitives or phoneme hypotheses from an audio signal [Usage] • Implement a classification algorithm that segments input percepts into output categories and quantitatively evaluates the resulting classification [Usage] • Evaluate the performance of the underlying feature-extraction, relative to at least one alternative possible approach (whether implemented or not) in its contribution to the classification task (8), above [Usage] • Describe at least three classification approaches, their prerequisites for applicability, their strengths, and their shortcomings [Usage]	• Computer vision – Image acquisition, representation, processing and properties – Shape representation, object recognition and segmentation – Motion analysis • Audio and speech recognition • Modularity in recognition • Approaches to pattern recognition – Classification algorithms and measures of classification quality – Statistical techniques
Readings : [Nil01], [RN03], [Pon+14]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS2902. Software Engineering II (Elective)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. Bibliography

[Amb01] Vincenzo Ambriola. *Software Process Technology.* Springer, July 2001.

[Blu92] Bruce I. Blum. *Software Engineering: A Holistic View.* 7th. Oxford University Press US, May 1992.

[Con00] R Conradi. *Software Process Technology.* Springer, Mar. 2000.

[Key04] Jessica Keyes. *Software Configuration Management.* CRC Press, Feb. 2004.

[Mon96] Carlo Montangero. *Software Process Technology.* Springer, Sept. 1996.

[Oqu03] Flavio Oquendo. *Software Process Technology.* Springer, Sept. 2003.

[Pre04] Roger S. Pressman. *Software Engineering: A Practitioner's Approach.* 6th. McGraw-Hill, Mar. 2004.

[PS01] John W. Priest and Jose M. Sanchez. *Product Development and Design for Manufacturing.* Marcel Dekker, Jan. 2001.

[Sch04] Stephen R Schach. *Object-Oriented and Classical Software Engineering.* McGraw-Hill, Jan. 2004.

[WA02] Daniel R. Windle and L. Rene Abreo. *Software Requirements Using the Unified Process.* Prentice Hall, Aug. 2002.

[WK00] Yingxu Wang and Graham King. *Software Engineering Processes: Principles and Applications.* CRC Press, Apr. 2000.

6. Information about the course

(a) **Brief description about the course** The topics of this course extend the ideas of software design and development from the introduction sequence to programming to encompass the problems encountered in large-scale projects. It is a broader and more complete view of Software Engineering appreciated from a Project point of view.

(b) **Prerrequisites:** CS2901. Software Engineering I. (5^{th} Sem)

(c) **Type of Course:** Elective

(d) **Modality:** Face to face

7. Specific goals of the Course

- Enable students to be part of and define software development teams facing real-world problems.
- familiarize the students with the process of administering a software project in such a way as to be able to create, improve and use tools and metrics that allow them to carry out the estimation and monitoring of a software project
- Create, evaluate and execute a test plan for medium-sized code segments, Distinguish between different types of tests, lay the foundation for creating, improve test procedures and tools for these purposes
- Select with justification an appropriate set of tools to support the development of a range of software products.

- Create, improve and use existing patterns for software maintenance. Disclose features and design patterns for software reuse.
- Identify and discuss different specialized systems, create, improve and use specialized standards for the design, implementation, maintenance and testing of specialized systems.

8. Contribution to Outcomes

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

f) An ability to communicate effectively. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

f) An ability to communicate effectively. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

9. Competences (IEEE)

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome c**

C11. Understanding of the concept of the lifecycle, including the significance of its phases (planning, development, deployment, and evolution).⇒ **Outcome c**

C12. Understanding the lifecycle implications for the development of all aspects of computer-related systems (including software, hardware, and human computer interface).⇒ **Outcome c,i**

C18. Ability to participate actively and as a member of a team. .⇒ **Outcome f**

CS1. Model and design computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.⇒ **Outcome c**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome c**

CS4. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.⇒ **Outcome c,i**

CS5. Specify, design, and implement computer-based systems.⇒ **Outcome c,i**

CS10. Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems. This should include tools for software control including version control and configuration management.⇒ **Outcome i**

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome c**

C11. Understanding of the concept of the lifecycle, including the significance of its phases (planning, development, deployment, and evolution).⇒ **Outcome c**

C12. Understanding the lifecycle implications for the development of all aspects of computer-related systems (including software, hardware, and human computer interface).⇒ **Outcome c,i**

C18. Ability to participate actively and as a member of a team. .⇒ **Outcome f**

CS1. Model and design computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.⇒ **Outcome c**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome c**

CS4. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.⇒ **Outcome c,i**

CS5. Specify, design, and implement computer-based systems.⇒ **Outcome c,i**

CS10. Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems. This should include tools for software control including version control and configuration management.⇒ **Outcome i**

10. List of topics

1. Tools and Environments
2. Software Verification and Validation
3. Software Evolution
4. Software Project Management

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Tools and Environments (12)	
Competences Expected: C1	
Learning Outcomes	**Topics**
• Software configuration management and version control [Usage] • Release management [Usage] • Requierements analysis and desing modeling tools [Usage] • Testing tools including static and dynamic analysis tools [Usage] • Programming enviroments that automate parts of program construction pocesses (e.g., automated builds) – Continuous integration [Usage] • Tool integration concepts and mechanisms [Usage]	• Software configuration management and version control • Release management • Requierements analysis and desing modeling tools • Testing tools including static and dynamic analysis tools • Programming enviroments that automate parts of program construction pocesses (e.g., automated builds) – Continuous integration • Tool integration concepts and mechanisms
Readings : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unit 2: Software Verification and Validation (12)	
Competences Expected: C20	
Learning Outcomes	**Topics**
• Distinguish between program validation and verification [Usage] • Describe the role that tools can play in the validation of software [Usage] • Undertake, as part of a team activity, an inspection of a medium-size code segment [Usage] • Describe and distinguish among the different types and levels of testing (unit, integration, systems, and acceptance) [Usage] • Describe techniques for identifying significant test cases for integration, regression and system testing [Usage] • Create and document a set of tests for a medium-size code segment [Usage] • Describe how to select good regression tests and automate them [Usage] • Use a defect tracking tool to manage software defects in a small software project [Usage] • Discuss the limitations of testing in a particular domain [Usage] • Evaluate a test suite for a medium-size code segment [Usage] • Compare static and dynamic approaches to verification [Usage] • Identify the fundamental principles of test-driven development methods and explain the role of automated testing in these methods [Usage] • Discuss the issues involving the testing of object-oriented software [Usage] • Describe techniques for the verification and validation of non-code artifacts [Usage] • Describe approaches for fault estimation [Usage] • Estimate the number of faults in a small software application based on fault density and fault seeding [Usage] • Conduct an inspection or review of software source code for a small or medium sized software project [Usage]	• Verification and validation concepts • Inspections, reviews, audits • Testing types, including human computer interface, usability, reliability, security, conformance to specification • Testing fundamentals – Unit, integration, validation, and system testing – Test plan creation and test case generation – Black-box and white-box testing techniques – Regression testing and test automation • Defect tracking • Limitations of testing in particular domains, such as parallel or safety-critical systems • Static approaches and dynamic approaches to verification • Test-driven development • Validation planning; documentation for validation • Object-oriented testing; systems testing • Verification and validation of non-code artifacts (documentation, help files, training materials) • Fault logging, fault tracking and technical support for such activities • Fault estimation and testing termination including defect seeding
Readings : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unit 3: Software Evolution (12)	
Competences Expected: C20	
Learning Outcomes	**Topics**
• Identify the principal issues associated with software evolution and explain their impact on the software lifecycle [Usage] • Estimate the impact of a change request to an existing product of medium size [Usage] • Use refactoring in the process of modifying a software component [Usage] • Discuss the challenges of evolving systems in a changing environment [Usage] • Outline the process of regression testing and its role in release management [Usage] • Discuss the advantages and disadvantages of different types of software reuse [Usage]	• Software development in the context of large, pre-existing code bases – Software change – Concerns and concernlocation – Refactoring • Software evolution • Characteristics of maintainable software • Reengineering systems • Software reuse – Code segments – Libraries and frameworks – Components – Product lines
Readings : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unit 4: Software Project Management (12)	
Competences Expected: C24	
Learning Outcomes	**Topics**
• Discuss common behaviors that contribute to the effective functioning of a team [Usage] • Create and follow an agenda for a team meeting [Usage] • Identify and justify necessary roles in a software development team [Usage] • Understand the sources, hazards, and potential benefits of team conflict [Usage] • Apply a conflict resolution strategy in a team setting [Usage] • Use an ad hoc method to estimate software development effort (eg, time) and compare to actual effort required [Usage] • List several examples of software risks [Usage] • Describe the impact of risk in a software development lifecycle [Usage] • Describe different categories of risk in software systems [Usage] • Demonstrate through involvement in a team project the central elements of team building and team management [Usage]	• Team participation – Team processes including responsabilities for task, meeting structure, and work schedule – Roles and responsabilities in a software team – Team conflict resolution – Risks associated with virtual teams (communication, perception, structure) • Effort estimation (at the personal level) • Risk – The role of risk in the lifecycle – Risk categories including security, safety, market, financial, technology, people, quality, structure and process • Team management – Team organization and decision-making – Role identification and assigment – Individual and team performance assessment • Project management – Scheduling and tracking – Project management tools – Cost/benefit analysis • Software measurement and estimation techniques • Software quality assurance and the role of measurements • Risk – Risk identification and management – Risk analysis and evaluation – Risk tolerance (e.g., risk-adverse, risk-neutral, risk-seeking) – Risk planning • System-wide approach to risk including hazards associated with tools
Readings : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS4002. Capstone Project I (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 3 HT; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Ass08] Association for Computing Machinery. *Digital Libray.* http://portal.acm.org/dl.cfm. Association for Computing Machinery, 2008.

[Cit08] CiteSeer.IST. *Scientific Literature Digital Libray.* http://citeseer.ist.psu.edu. College of Information Sciences and Technology, Penn State University, 2008.

[IEE08] IEEE-Computer Society. *Digital Libray.* http://www.computer.org/publications/dlib. IEEE-Computer Society, 2008.

6. **Information about the course**

(a) **Brief description about the course** This course aims to allow the student to carry out a study of the state of the art of a topic chosen by the student for his thesis.

(b) **Prerrequisites:** CS2102. Analysis and Design of Algorithms. (5^{th} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- That the student carries out an initial investigation in a specific subject realizing the study of the state of the art of the chosen subject.
- That the student shows mastery in the subject of the line of investigation chosen
- That the student choose a teacher who dominates the research chosen as an advisor.
- The deliverables of this course are:

  Avance parcial: Solid bibliography and progress of a Technical Reporto.

  Final: Technical Report with preliminary comparative experiments that demonstrate that the student already knows the existing techniques in the area of his project and choose a teacher who dominates the area of his project as an adviser of his project.

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Assessment**)

f) An ability to communicate effectively. (**Usage**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

l) Develop principles research in the area of computing with levels of international competitiveness. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Assessment**)

f) An ability to communicate effectively. (**Usage**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

l) Develop principles research in the area of computing with levels of international competitiveness. (**Usage**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a,b,c**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome e,f,g**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome h,i,l**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a,b,c**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome e,f,g**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome h,i,l**

10. List of topics

1. Lifting the state of the art

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

<table>
<tr><th colspan="2">Unit 1: Lifting the state of the art (60)</th></tr>
<tr><th colspan="2">Competences Expected: C1,C20,CS2</th></tr>
<tr><th>Learning Outcomes</th><th>Topics</th></tr>
<tr><td>• Make a bibliographical survey of the state of the art of the chosen subject (this probably means 1 or 2 chapters of theoretical framework in addition to the introduction that is chapter I of the thesis) [Usage]
• Writing a latex document in paper format with higher quality than Project I (master tables, figures, equations, indices, bibtex, cross references, citations, pstricks) [Usage]
• Try to make presentations using prosper [Usage]
• Show basic experiments [Usage]
• Choose an advisor who dominates the research area [Usage]</td><td>• Perform an in-depth study of the state of the art in a certain topic in the area of Computation.
• Writing technical articles in computing.</td></tr>
<tr><td colspan="2"><b>Readings :</b> [IEE08], [Ass08], [Cit08]</td></tr>
</table>

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** ET201. Entrerpreneurship I (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 2 HT; 2 HP; (15 weeks)
4. **Professor(s)**

   Meetings after coordination with the professor
5. **Bibliography**

[BD12] Steve Blank and Bob Dorf. *The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company*. K and S Ranch, 2012.

[BDN10] Thomas Byers, Richard Dorf, and Andrew Nelson. *Technology Ventures: From Idea to Enterprise*. McGraw-Hill Science, 2010.

[Con96] Congreso de la Republica del Perú. *Decreto Legislativo Nº823. Ley de la Propiedad Industrial*. El Peruano, 1996.

[Gar+14] René Garzozi-Pincay et al. *Planes de Negocios para Emprendedores*. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.

[OP10] Alexander Osterwalder and Yves Pigneur. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Wiley, 2010.

[Rep97] Congreso de la Republica del Peru. *Ley Nº26887. Ley General de Sociedades*. El Peruano, 1997.

[Rie11] Eric Ries. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business, 2011.

6. **Information about the course**

   (a) **Brief description about the course** Este es el primer curso dentro del área de formación de empresas de base tecnológica, tiene como objetivo dotar al futuro profesional de conocimientos, actitudes y aptitudes que le permitan elaborar un plan de negocio para una empresa de base tecnológica. El curso está dividido en las siguientes unidades: Introducción, Creatividad, De la idea a la oportunidad, el modelo Canvas, Customer Development y Lean Startup, Aspectos Legales y Marketing, Finanzas de la empresa y Presentación.

   Se busca aprovechar el potencial creativo e innovador y el esfuerzo de los alumnos en la creación de nuevas empresas.

   (b) **Prerrequisites:** None

   (c) **Type of Course:** Mandatory

   (d) **Modality:** Face to face

7. **Specific goals of the Course**

   - Que el alumno conozca como elaborar un plan de negocio para dar inicio a una empresa de base tecnológica.
   - Que el alumno sea capaz de realizar, usando modelos de negocio, la concepción y presentación de una propuesta de negocio.

8. **Contribution to Outcomes**

d) An ability to function on multidisciplinary teams. (**Usage**)

f) An ability to communicate effectively. (**Assessment**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

k) Apply the principles of development and design in the construction of software systems of variable complexity. (**Assessment**)

m) Transform your knowledge of the area of Computer Science into technological enterprises. (**Assessment**)

9. Competences (IEEE)

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome d**

C10. Understanding of the impact on individuals, organizations, and society of deploying technological solutions and interventions.⇒ **Outcome f**

C17. Ability to properly express in oral and written media as expected from a university graduate. ⇒ **Outcome f**

C18. Ability to participate actively and as a member of a team. .⇒ **Outcome i**

C19. Ability to effectively identify the goals and priorities of their project, stating the action, the time and resources required.⇒ **Outcome i**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome k**

C23. Ability to undertake, complete, and present a capstone project.⇒ **Outcome k**

CS5. Specify, design, and implement computer-based systems.⇒ **Outcome m**

10. List of topics

1.
2.
3.
4.
5.
6.
7.
8.

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: (5)	
Competences Expected: C2	
Learning Outcomes	**Topics**
• Identificar características de los emprendedores [Familiarity] • Introducir modelos de negocio [Familiarity]	• Emprendedor, emprendedurismo e innovación tecnológica • Modelos de negocio • Formación de equipos
Readings : [BDN10], [OP10], [Gar+14]	

Unit 2: (5)	
Competences Expected: C10	
Learning Outcomes	**Topics**
• Plantear correctamente la vision y misión de empresa [Usage] • Caracterizar una propuesta de valor innovadora [Assessment] • Identificar los diversos tipos y fuentes de innovación [Familiarity]	• Visión • Misión • La Propuesta de valor • Creatividad e invención • Tipos y fuentes de innovación • Estrategia y Tecnología • Escala y ámbito
Readings : [BDN10], [BD12], [Gar+14]	

Unit 3: (5)	
Competences Expected: C17	
Learning Outcomes	**Topics**
• Conocer estrategias empresariales [Familiarity] • Caracterizar barreras y ventajas competitivas [Familiarity]	• Estrategia de la Empresa • Barreras • Ventaja competitiva sostenible • Alianzas • Aprendizaje organizacional • Desarrollo y diseño de productos
Readings : [BDN10], [OP10], [Rie11], [Gar+14]	

Unit 4: (20)	
Competences Expected: C18	
Learning Outcomes	**Topics**
• Conocer los elementos del modelo Canvas [Usage] • Elaborar un plan de negocio basado en el modelo Canvas [Usage]	• Creación de un nuevo negocio • El plan de negocio • Canvas • Elementos del Canvas
Readings : [OP10], [BD12], [Gar+14]	

Unit 5: (20)	
Competences Expected: C19	
Learning Outcomes	**Topics**
• Conocer y aplicar el modelo Customer Development [Usage] • Conocer y aplicar el modelo Lean Startup [Usage]	• Aceleración versus incubación • Customer Development • Lean Startup
Readings : [BD12], [Rie11], [Gar+14]	

Unit 6: (5)	
Competences Expected: C20	
Learning Outcomes	**Topics**
• Conocer los aspectos legales necesarios para la formación de una empresa tecnológica [Familiarity] • Identificar segmentos de mercado y objetivos de marketing [Familiarity]	• Aspectos Legales y tributarios para la constitución de la empresa • Propiedad intelectual • Patentes • Copyrights y marca registrada • Objetivos de marketing y segmentos de mercado • Investigación de mercado y búsqueda de clientes
Readings : [BDN10], [Rie11], [Con96], [Rep97], [Gar+14]	

Unit 7: (5)	
Competences Expected: C23	
Learning Outcomes	**Topics**
• Definir um modelo de costos y utilidades [Assessment] • Conocer las diversas fuentes de financiamento [Familiarity]	• Modelo de costos • Modelo de utilidades • Precio • Plan financiero • Formas de financiamiento • Fuentes de capital • Capital de riesgo
Readings : [BDN10], [BD12], [Gar+14]	

Unit 8: (5)	
Competences Expected: CS5	
Learning Outcomes	**Topics**
• Conocer las diversas formas de presentar propuestas de negocio [Familiarity] • Realizar la presentación de una propuesta de negocio [Usage]	• The Elevator Pitch • Presentación • Negociación
Readings : [BDN10], [BD12], [Gar+14]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS3700. Big Data (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 1 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Bal+08] Shumeet Baluja et al. "Video Suggestion and Discovery for Youtube: Taking Random Walks Through the View Graph". In: *Proceedings of the 17th International Conference on World Wide Web*. WWW '08. Beijing, China: ACM, 2008, pp. 895–904. ISBN: 978-1-60558-085-2. DOI: 10.1145/1367497.1367618. URL: http://doi.acm.org/10.1145/1367497.1367618.

[BVS13] Rajkumar Buyya, Christian Vecchiola, and S. Thamarai Selvi. *Mastering Cloud Computing: Foundations and Applications Programming*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013. ISBN: 9780124095397, 9780124114548.

[Cou+11] George Coulouris et al. *Distributed Systems: Concepts and Design*. 5th. USA: Addison-Wesley Publishing Company, 2011. ISBN: 0132143011, 9780132143011.

[HDF11] Kai Hwang, Jack Dongarra, and Geoffrey C. Fox. *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN: 0123858801, 9780123858801.

[Low+12] Yucheng Low et al. "Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud". In: *Proc. VLDB Endow.* 5.8 (Apr. 2012), pp. 716–727. ISSN: 2150-8097. DOI: 10.14778/2212351.2212354. URL: http://dx.doi.org/10.14778/2212351.2212354.

[Mal+10] Grzegorz Malewicz et al. "Pregel: A System for Large-scale Graph Processing". In: *ACM SIGMOD Record*. SIGMOD '10 (2010), pp. 135–146. DOI: 10.1145/1807167.1807184. URL: http://doi.acm.org/10.1145/1807167.1807184.

6. **Information about the course**

(a) **Brief description about the course** Nowadays, knowing scalable approaches to processing and storing large volumes of information (terabytes, petabytes and even exabytes) is fundamental in computer science courses. Every day, every hour, every minute generates a large amount of information which needs to be processed, stored, analyzed.

(b) **Prerrequisites:**

- CS2702. Databases II. (5th Sem)
- CS3P01. Parallel and Distributed Computing . (7th Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- That the student is able to create parallel applications to process large volumes of information
- That the student is able to compare the alternatives for the processing of big data

- That the student is able to propose architectures for a scalable application

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

9. Competences (IEEE)

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a,b**

C16. Ability to identify advanced computing topics and understanding the frontiers of the discipline.⇒ **Outcome i**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome i,b**

CS3. Analyze the extent to which a computer-based system meets the criteria defined for its current use and future development.⇒ **Outcome j**

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome j**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a,b**

C16. Ability to identify advanced computing topics and understanding the frontiers of the discipline.⇒ **Outcome i**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome i,b**

CS3. Analyze the extent to which a computer-based system meets the criteria defined for its current use and future development.⇒ **Outcome j**

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome j**

10. List of topics

1. Introducción a Big Data

2. Hadoop

3. Procesamiento de Grafos en larga escala

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Introducción a Big Data (15)	
Competences Expected: C2, C4	
Learning Outcomes	**Topics**
- Explain the concept of Cloud Computing from the point of view of Big Data[Familiarity] - Explain the concept of Distributed File System [Familiarity] - Explain the concept of the MapReduce programming model[Familiarity]	- Overview on Cloud Computing - Distributed File System Overview - Overview of the MapReduce programming model
Readings : [Cou+11]	

Unit 2: Hadoop (15)	
Competences Expected: C2, C4	
Learning Outcomes	**Topics**
- Understand and explain the Hadoop suite [Familiarity] - Implement solutions using the MapReduce programming model. [Usage] - Understand how data is saved in the HDFS. [Familiarity]	- Hadoop overview. - History. - Hadoop Structure. - HDFS, Hadoop Distributed File System. - Programming Model MapReduce
Readings : [HDF11], [BVS13]	

Unit 3: Procesamiento de Grafos en larga escala (10)	
Competences Expected: C16	
Learning Outcomes	**Topics**
• Understand and explain the architecture of the Pregel project. [Familiarity] • Understand the GraphLab project architecture. [Familiarity] • Understand the architecture of the Giraph project. [Familiarity] • Implement solutions using Pregel, GraphLab or Giraph. [Usage]	• Pregel: A System for Large-scale Graph Processing. • Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud. • Apache Giraph is an iterative graph processing system built for high scalability.
Readings : [Low+12], [Mal+10], [Bal+08]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS3I01. Computer Security (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 1 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[WL14] Stallings. W and Brown. L. *Computer Security: Principles and Practice*. Pearson Education, Limited, 2014. ISBN: 9780133773927.

6. **Information about the course**

(a) **Brief description about the course** Nowadays, information is one of the most valuable assets in any organization. This course is oriented to be able to provide the student with the security elements oriented to protect the Information of the organization and mainly to be able to foresee the possible problems related to this heading. This subject involves the development of a preventive attitude on the part of the student in all areas related to software development.

(b) **Prerrequisites:** CS2301. Networking and Communication. (6^{th} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Discuss at an intermediate intermediate level the fundamentals of Computer Security.
- Provide different aspects of the malicious code.
- That the student knows the concepts of cryptography and security in computer networks.
- Discuss and analyze together with the student the aspects of Internet Security.

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Assessment**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Assessment**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

9. Competences (IEEE)

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome g**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome g,a**

C21. Understanding the professional, legal, security, political, humanistic, environmental, cultural and ethical issues. ⇒ **Outcome e**

C22. Ability to demonstrate attitudes and priorities that honor, protect, and enhance the profession's ethical stature and standing..⇒ **Outcome e,h**

CS7. Apply the principles of effective information management, information organization, and information-retrieval skills to information of various kinds, including text, images, sound, and video. This must include managing any security issues.⇒ **Outcome i,h**

CS9. Identify any risks (and this includes any safety or security aspects) that may be involved in the operation of computing equipment within a given context. ⇒ **Outcome b**

CS11. Be aware of the existence of publicly available software and understanding the potential of opensource projects.⇒ **Outcome b,g**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome g**

C9. Understanding of computing's limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.⇒ **Outcome g,a**

C21. Understanding the professional, legal, security, political, humanistic, environmental, cultural and ethical issues. ⇒ **Outcome e**

C22. Ability to demonstrate attitudes and priorities that honor, protect, and enhance the profession's ethical stature and standing..⇒ **Outcome e,h**

CS7. Apply the principles of effective information management, information organization, and information-retrieval skills to information of various kinds, including text, images, sound, and video. This must include managing any security issues.⇒ **Outcome i,h**

CS9. Identify any risks (and this includes any safety or security aspects) that may be involved in the operation of computing equipment within a given context. ⇒ **Outcome b**

CS11. Be aware of the existence of publicly available software and understanding the potential of opensource projects.⇒ **Outcome b,g**

10. List of topics

1. Foundational Concepts in Security
2. Principles of Secure Design
3. Defensive Programming

4. Threats and Attacks
5. Network Security
6. Cryptography
7. Web Security
8. Platform Security
9. Digital Forensics
10. Secure Software Engineering

11. Methodology and Evaluation
Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

Unit 1: Foundational Concepts in Security (25)	
Competences Expected: C2,C8	
Learning Outcomes	**Topics**
• Analyze the tradeoffs of balancing key security properties (Confidentiality, Integrity, Availability) [Familiarity] • Describe the concepts of risk, threats, vulnerabilities and attack vectors (including the fact that there is no such thing as perfect security) [Familiarity] • Explain the concepts of authentication, authorization, access control [Familiarity] • Explain the concept of trust and trustworthiness [Familiarity] • Recognize that there are important ethical issues to consider in computer security, including ethical issues associated with fixing or not fixing vulnerabilities and disclosing or not disclosing vulnerabilities [Familiarity]	• CIA (Confidentiality, Integrity, Availability) • Concepts of risk, threats, vulnerabilities, and attack vectors • Authentication and authorization, access control (mandatory vs. discretionary) • Concept of trust and trustworthiness • Ethics (responsible disclosure)
Readings : [WL14]	

Unit 2: Principles of Secure Design (25)	
Competences Expected: C,9C21,C22	
Learning Outcomes	**Topics**
• Describe the principle of least privilege and isolation as applied to system design [Familiarity] • Summarize the principle of fail-safe and deny-by-default [Familiarity] • Discuss the implications of relying on open design or the secrecy of design for security. [Familiarity] • Explain the goals of end-to-end data security [Familiarity] • Discuss the benefits of having multiple layers of defenses [Familiarity] • For each stage in the lifecycle of a product, describe what security considerations should be evaluated. [Familiarity] • Describe the cost and tradeoffs associated with designing security into a product [Familiarity] • Describe the concept of mediation and the principle of complete mediation [Familiarity] • Be aware of standard components for security operations, instead of re-inventing fundamentals operations [Familiarity] • Explain the concept of trusted computing including trusted computing base and attack surface and the principle of minimizing trusted computing base [Familiarity] • Discuss the importance of usability in security mechanism design [Familiarity] • Describe security issues that arise at boundaries between multiple components. [Familiarity] • Identify the different roles of prevention mechanisms and detection/deterrence mechanisms [Familiarity]	• Least privilege and isolation • Fail-safe defaults • Open design • End-to-end security • Defense in depth (e.g., defensive programming, layered defense) • Security by design • Tensions between security and other design goals • Complete mediation • Use of vetted security components • Economy of mechanism (reducing trusted computing base, minimize attack surface) • Usable security • Security composability • Prevention, detection, and deterrence
Readings : [WL14]	

Unit 3: Defensive Programming (25)

Competences Expected: CS6,CS7,CS9

Learning Outcomes	**Topics**
• Explain why input validation and data sanitization is necessary in the face of adversarial control of the input channel. [Usage] • Explain why you might choose to develop a program in a type-safe language like Java, in contrast to an unsafe programming language like C/C++ [Usage] • Classify common input validation errors, and write correct input validation code [Usage] • Demonstrate using a high-level programming language how to prevent a race condition from occurring and how to handle an exception [Usage] • Demonstrate the identification and graceful handling of error conditions [Familiarity] • Explain the risks with misusing interfaces with third-party code and how to correctly use third-party code [Familiarity] • Discuss the need to update software to fix security vulnerabilities and the lifecycle management of the fix [Familiarity]	• Input validation and data sanitization • Choice of programming language and type-safe languages • Examples of input validation and data sanitization errors – Buffer overflows – Integer errors – SQL injection – XSS vulnerability • Race conditions • Correct handling of exceptions and unexpected behaviors • Correct usage of third-party components • Effectively deploying security updates • Information flow control • Correctly generating randomness for security purposes • Mechanisms for detecting and mitigating input and data sanitization errors • Fuzzing • Static analysis and dynamic analysis • Program verification • Operating system support (e.g., address space randomization, canaries) • Hardware support (e.g, DEP, TPM)

Readings : [WL14]

Unit 4: Threats and Attacks (25)	
Competences Expected: CS6,CS7,CS9	
Learning Outcomes	**Topics**
• Describe likely attacker types against a particular system [Familiarity] • Discuss the limitations of malware countermeasures (eg, signature-based detection, behavioral detection) [Familiarity] • Identify instances of social engineering attacks and Denial of Service attacks [Familiarity] • Discuss how Denial of Service attacks can be identified and mitigated [Familiarity] • Describe risks to privacy and anonymity in commonly used applications [Familiarity] • Discuss the concepts of covert channels and other data leakage procedures [Familiarity]	• Attacker goals, capabilities, and motivations (such as underground economy, digital espionage, cyberwarfare, insider threats, hacktivism, advanced persistent threats) • Examples of malware (e.g., viruses, worms, spyware, botnets, Trojan horses or rootkits) • Denial of Service (DoS) and Distributed Denial of Service (DDoS) • Social engineering (e.g., phishing) • Attacks on privacy and anonymity • Malware/unwanted communication such as covert channels and steganography
Readings : [WL14]	

Unit 5: Network Security (25)	
Competences Expected: CS6,CS7,CS9	
Learning Outcomes	**Topics**
• Describe the different categories of network threats and attacks [Familiarity] • Describe the architecture for public and private key cryptography and how PKI supports network security [Familiarity] • Describe virtues and limitations of security technologies at each layer of the network stack [Familiarity] • Identify the appropriate defense mechanism(s) and its limitations given a network threat [Usage]	• Network specific threats and attack types (e.g., denial of service, spoofing, sniffing and traffic redirection, man-in-the-middle, message integrity attacks, routing attacks, and traffic analysis) • Use of cryptography for data and network security • Architectures for secure networks (e.g., secure channels, secure routing protocols, secure DNS, VPNs, anonymous communication protocols, isolation) • Defense mechanisms and countermeasures (e.g., network monitoring, intrusion detection, firewalls, spoofing and DoS protection, honeypots, tracebacks) • Security for wireless, cellular networks • Other non-wired networks (e.g., ad hoc, sensor, and vehicular networks) • Censorship resistance • Operational network security management (e.g., configure network access control)
Readings : [WL14]	

Unit 6: Cryptography (25)	
Competences Expected: CS6,CS7,CS9	
Learning Outcomes	**Topics**
• Describe the purpose of Cryptography and list ways it is used in data communications [Familiarity] • Define the following terms: Cipher, Cryptanalysis, Cryptographic Algorithm, and Cryptology and describe the two basic methods (ciphers) for transforming plain text in cipher text [Familiarity] • Discuss the importance of prime numbers in cryptography and explain their use in cryptographic algorithms [Familiarity] • Illustrate how to measure entropy and how to generate cryptographic randomness [Usage] • Use public-key primitives and their applications [Usage] • Explain how key exchange protocols work and how they fail [Familiarity] • Discuss cryptographic protocols and their properties [Familiarity]	• Basic Cryptography Terminology covering notions pertaining to the different (communication) partners, secure/unsecure channel, attackers and their capabilities, encryption, decryption, keys and their characteristics, signatures • Cipher types (e.g., Caesar cipher, affine cipher) together with typical attack methods such as frequency analysis • Public Key Infrastructure support for digital signature and encryption and its challenges • Symmetric key cryptography – Perfect secrecy and the one time pad – Modes of operation for semantic security and authenticated encryption (e.g., encrypt-then-MAC, OCB, GCM) – Message integrity (e.g., CMAC, HMAC) • Public key cryptography: – Trapdoor permutation, e.g., RSA – Public key encryption, e.g., RSA encryption, EI Gamal encryption – Digital signatures – Public-key infrastructure (PKI) and certificates – Hardness assumptions, e.g., Diffie-Hellman, integer factoring • Authenticated key exchange protocols, e.g., TLS • Cryptographic primitives: – pseudo-random generators and stream ciphers – block ciphers (pseudo-random permutations), e.g., AES – pseudo-random functions – hash functions, e.g., SHA2, collision resistance – message authentication codes – key derivations functions
Readings : [WL14]	

Unit 7: Web Security (25)	
Competences Expected: C8,C9	
Learning Outcomes	**Topics**
• Describe the browser security model including same-origin policy and threat models in web security [Familiarity] • Discuss the concept of web sessions, secure communication channels such as TLS and importance of secure certificates, authentication including single sign-on such as OAuth and SAML [Familiarity] • Investigate common types of vulnerabilities and attacks in web applications, and defenses against them [Familiarity] • Use client-side security capabilities [Usage]	• Web security model – Browser security model including same-origin policy – Client-server trust boundaries, e.g., cannot rely on secure execution in the client • Session management, authentication – Single sign-on – HTTPS and certificates • Application vulnerabilities and defenses – SQL injection – XSS – CSRF • Client-side security – Cookies security policy – HTTP security extensions, e.g. HSTS – Plugins, extensions, and web apps – Web user tracking • Server-side security tools, e.g. Web Application Firewalls (WAFs) and fuzzers
Readings : [WL14]	

Unit 8: Platform Security (25)	
Competences Expected: CS6,CS7,CS9	
Learning Outcomes	**Topics**
• Explain the concept of code integrity and code signing and the scope it applies to [Familiarity] • Discuss the concept of root of trust and the process of secure boot and secure loading [Familiarity] • Describe the mechanism of remote attestation of system integrity [Familiarity] • Summarize the goals and key primitives of TPM [Familiarity] • Identify the threats of plugging peripherals into a device [Familiarity] • Identify physical attacks and countermeasures [Familiarity] • Identify attacks on non-PC hardware platforms [Familiarity] • Discuss the concept and importance of trusted path [Familiarity]	• Code integrity and code signing • Secure boot, measured boot, and root of trust • Attestation • TPM and secure co-processors • Security threats from peripherals, e.g., DMA, IOMMU • Physical attacks: hardware Trojans, memory probes, cold boot attacks • Security of embedded devices, e.g., medical devices, cars • Trusted path
Readings : [WL14]	

Unit 9: Digital Forensics (25)	
Competences Expected: C8,C9	
Learning Outcomes	**Topics**
• Describe what is a Digital Investigation is, the sources of digital evidence, and the limitations of forensics [Familiarity] • Explain how to design software to support forensics [Familiarity] • Describe the legal requirements for use of seized data [Familiarity] • Describe the process of evidence seizure from the time when the requirement was identified to the disposition of the data [Familiarity] • Describe how data collection is accomplished and the proper storage of the original and forensics copy [Familiarity] • Conduct data collection on a hard drive [Usage] • Describe a person's responsibility and liability while testifying as a forensics examiner [Familiarity] • Recover data based on a given search term from an imaged system [Usage] • Reconstruct application history from application artifacts [Familiarity] • Reconstruct web browsing history from web artifacts [Familiarity] • Capture and interpret network traffic [Familiarity] • Discuss the challenges associated with mobile device forensics [Familiarity]	• Basic Principles and methodologies for digital forensics • Design systems with forensic needs in mind • Rules of Evidence - general concepts and differences between jurisdictions and Chain of Custody • Search and Seizure of evidence: legal and procedural requirements • Digital Evidence methods and standards • Techniques and standards for Preservation of Data • Legal and Reporting Issues including working as an expert witness • OS/File System Forensics • Application Forensics • Web Forensics • Network Forensics • Mobile Device Forensics • Computer/network/system attacks • Attack detection and investigation • Anti-forensics
Readings : [WL14]	

Unit 10: Secure Software Engineering (25)	
Competences Expected: C21,C22	
Learning Outcomes	**Topics**
• Describe the requirements for integrating security into the SDL [Familiarity] • Apply the concepts of the Design Principles for Protection Mechanisms, the Principles for Software Security (Viega and McGraw), and the Principles for Secure Design (Morrie Gasser) on a software development project [Familiarity] • Develop specifications for a software development effort that fully specify functional requirements and identifies the expected execution paths [Familiarity]	• Building security into the software development lifecycle • Secure design principles and patterns • Secure software specifications and requirements • Secure software development practices • Secure testing- the process of testing that security requirements are met (including static and dynamic analysis).
Readings : [WL14]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS4003. Final Project II (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HP; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Ass08] Association for Computing Machinery. *Digital Libray*. http://portal.acm.org/dl.cfm. Association for Computing Machinery, 2008.

[Cit08] CiteSeer.IST. *Scientific Literature Digital Libray*. http://citeseer.ist.psu.edu. College of Information Sciences and Technology, Penn State University, 2008.

[IEE08] IEEE-Computer Society. *Digital Libray*. http://www.computer.org/publications/dlib. IEEE-Computer Society, 2008.

6. Information about the course

(a) **Brief description about the course** This course aims at the student to conclude his thesis project.

(b) **Prerrequisites:** CS4002. Capstone Project I. (8th Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- That the student is in the capacity to formally present his thesis project with the theoretical framework and complete bibliographic survey.
- That the student master the state of the art of his area of research.
- The deliverables of this course are:

  Avance parcial: Thesis plan progress including motivation and context, problem definition, objectives, schedule of activities up to the final thesis project and the state of the art of the topic addressed.

  Final: Complete thesis plan and advancement of Thesis including theoretical framework chapters, related works and preliminary (formal or statistical) results oriented to your thesis topic.

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Assessment**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Assessment**)

f) An ability to communicate effectively. (**Assessment**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Assessment**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

l) Develop principles research in the area of computing with levels of international competitiveness. (**Assessment**)

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Assessment**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Assessment**)

f) An ability to communicate effectively. (**Assessment**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Assessment**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

l) Develop principles research in the area of computing with levels of international competitiveness. (**Assessment**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a,b,c**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome e,f.g**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome h,i,l**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a,b,c**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome e,f.g**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome h,i,l**

10. List of topics

1. Thesis project
2. Thesis progress

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is

considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

Unit 1: Thesis project (30)	
Competences Expected: C1,C20,CS2	
Learning Outcomes	**Topics**
- Description of the format used by the University for the thesis[Assessment] - Conclude the thesis project plan[Assessment] - Present the state of the art thesis topic(50%)[Assessment]	- Thesis project.
Readings : [IEE08], [Ass08], [Cit08]	

Unit 2: Thesis progress (30)	
Competences Expected: C1,C20,CS2	
Learning Outcomes	**Topics**
- Description of the format used by the University for the thesis[Assessment] - Conclude the chapter of the theoretical framework of the Thesis[Assessment] - Complete the chapter on related works(35%)[Assessment] - Plan, develop and present results (formal or statistical) of experiments oriented to your thesis topic (35%)[Assessment]	- Thesis Progress.
Readings : [IEE08], [Ass08], [Cit08]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS3501. Topics in Computer Graphics (Elective)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[HB90] Donald Hearn and Pauline Baker. *Computer Graphics in C*. Prentice Hall, 1990.

[Hug+13] John F. Hughes et al. *Computer Graphics - Principles and Practice 3rd Edition*. Addison-Wesley, 2013.

6. **Information about the course**

(a) **Brief description about the course** In this course you can delve into any of the topics Mentioned in the area of Graphics Computing (Graphics and Visual Computing - GV).

This course is designed to perform some advanced course suggested by the ACM / IEEE curriculum. [Hug+13; HB90]

(b) **Prerrequisites:** CS2501. Computer graphics . (7^{th} Sem)

(c) **Type of Course:** Elective

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- That the student uses computer techniques Graphs that involve complex data structures and algorithms.
- That the student apply the concepts learned to create an application about a real problem.
- That the student investigate the possibility of creating a new algorithm and / or new technique to solve a real problem

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

l) Develop principles research in the area of computing with levels of international competitiveness. (**Usage**)

m) Transform your knowledge of the area of Computer Science into technological enterprises. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

l) Develop principles research in the area of computing with levels of international competitiveness. (**Usage**)

m) Transform your knowledge of the area of Computer Science into technological enterprises. (**Usage**)

9. **Competences (IEEE)**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a,b**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome l,m**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a,b**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome l,m**

10. List of topics

1. Advanced Topics on Computer Graphics

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Advanced Topics on Computer Graphics (0)	
Competences Expected: 4	
Learning Outcomes	**Topics**
• Advanced Topics on Computer Graphics	• CS355. Advanced Computer Graphics • CS356. Computer animation • CS313. Geometric Algorithms • CS357. visualization • CS358. Virtual reality • CS359. Genetic algorithms
Readings : [**Soars022S**], [**Soars022W**], [**Soars022T**], [**Cambridge06**], [**MacGrew99**]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS3602. Robotics (Elective)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[MVR07] Sonka. M, Hlavac. V, and Boile. R. *Image Processing, Analysis and Machine Vision*. Cengage-Engineering, 2007.

[RR07] Gonzales. R C and Woods. R E. *Digital Image Processing*. Prentice Hall, 2007. ISBN: 013168728X,978013168728B.

[SN04] R. Siegwart and I. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. The MIT Press., 2004. ISBN: 0-262-19502-X.

[Sto00] Peter Stone. *Layered Learning in Multiagent Systems*. Intelligent Robots and Autonomous Agents. The MIT Press, 2000. ISBN: 9780262194389.

[SWD05] Thrun. S, Burgard. W, and Fox. D. *Probabilistic Robotics*. Intelligent Robots and Autonomous Agents. The MIT Press, 2005.

6. **Information about the course**

(a) **Brief description about the course** That the student knows and understands the concepts and fundamental principles of control, road planning and the definition of strategies in robotics as well as concepts of robotic perception in a way that understands the potential of robotic systems

(b) **Prerrequisites:** CS2601. Artificial intelligence . (7^{th} Sem)

(c) **Type of Course:** Elective

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Synthesize the potential and limitations of the state-of-the-art of today's robotic systems.
- Implement Simple Motion Planning Algorithms.
- Explain the uncertainties associated with sensors and how to treat them.
- Designing a Simple Control Architecture.
- Describes several navigation strategies
- Describe the importance of recognizing images and objects in intelligent systems
- Outline the main techniques of object recognition
- Describe the different characteristics of the technologies used in perception

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Usage**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Familiarity**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Usage**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Familiarity**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

9. Competences (IEEE)

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a,b**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome e**

C23. Ability to undertake, complete, and present a capstone project.⇒ **Outcome b,i,h**

CS1. Model and design computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.⇒ **Outcome b**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a,b**

C8. Understanding of what current technologies can and cannot accomplish. ⇒ **Outcome e**

C23. Ability to undertake, complete, and present a capstone project.⇒ **Outcome b,i,h**

CS1. Model and design computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.⇒ **Outcome b**

10. List of topics

1. Robotics
2. Robotics
3. Robotics
4. Perception and Computer Vision
5. Robotics

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

Unit 1: Robotics (5)	
Competences Expected: CS12	
Learning Outcomes	**Topics**
• List capabilities and limitations of today's state-of-the-art robot systems, including their sensors and the crucial sensor processing that informs those systems [Familiarity] • Integrate sensors, actuators, and software into a robot designed to undertake some task [Usage]	• Overview: problems and progress – State-of-the-art robot systems, including their sensors and an overview of their sensor processing – Robot control architectures, e.g., deliberative vs. reactive control and Braitenberg vehicles – World modeling and world models – Inherent uncertainty in sensing and in control • Configuration space and environmental maps
Readings : [SN04], [SWD05], [Sto00]	

Unit 2: Robotics (15)	
Competences Expected: C2,C23	
Learning Outcomes	**Topics**
• Program a robot to accomplish simple tasks using deliberative, reactive, and/or hybrid control architectures [Usage] • Implement fundamental motion planning algorithms within a robot configuration space [Usage]	• Interpreting uncertain sensor data • Localizing and mapping
Readings : [SN04], [SWD05]	

Unit 3: Robotics (20)	
Competences Expected: CS1	
Learning Outcomes	**Topics**
• Characterize the uncertainties associated with common robot sensors and actuators; articulate strategies for mitigating these uncertainties [Usage] • List the differences among robots' representations of their external environment, including their strengths and shortcomings [Usage]	• Navigation and control • Motion planning
Readings : [SN04]	

Unit 4: Perception and Computer Vision (10)	
Competences Expected: C2,CS1	
Learning Outcomes	**Topics**
• Summarize the importance of image and object recognition in AI and indicate several significant applications of this technology [Usage] • Implement 2d object recognition based on contour- and/or region-based shape representations [Usage]	• Computer vision – Image acquisition, representation, processing and properties – Shape representation, object recognition and segmentation – Motion analysis • Modularity in recognition
Readings : [MVR07], [RR07]	

Unit 5: Robotics (10)	
Competences Expected: C23,CS1	
Learning Outcomes	**Topics**
• Compare and contrast at least three strategies for robot navigation within known and/or unknown environments, including their strengths and shortcomings [Familiarity] • Describe at least one approach for coordinating the actions and sensing of several robots to accomplish a single task [Familiarity]	• Multiple-robot coordination
Readings : [Sto00]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS3901. Software Engineering III (Elective)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[PM14] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2014.

[Som10] Ian Sommerville. *Software Engineering*. 9th. Addison-Wesley, Mar. 2010.

6. **Information about the course**

(a) **Brief description about the course** Software development requires the use of best development practices, IT project management, equipment management And efficient and rational use of quality assurance frameworks, these elements are key and transversal during the whole productive process. The construction of software contemplates the implementation and use of processes, methods, models and tools that allow to achieve the realization of the quality attributes of a product.

(b) **Prerrequisites:** CS2902. Software Engineering II . (7^{th} Sem)

(c) **Type of Course:** Elective

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Understand and implement the fundamental concepts of project management and software equipment management.
- Understand the fundamentals of project management, including its definition, scope, and need for project management in the modern organization.
- Students have to understand the fundamental concepts of CMMI, PSP, TSP to be adopted in software projects.
- Describe and understand quality assurance models as a key framework for the success of IT projects.

8. **Contribution to Outcomes**

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

m) Transform your knowledge of the area of Computer Science into technological enterprises. (**Assessment**)

o) Improve the conditions of society by putting technology at the service of the human being. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

m) Transform your knowledge of the area of Computer Science into technological enterprises. (**Assessment**)

o) Improve the conditions of society by putting technology at the service of the human being. (**Usage**)

9. Competences (IEEE)

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome c**

C11. Understanding of the concept of the lifecycle, including the significance of its phases (planning, development, deployment, and evolution).⇒ **Outcome i,k**

C12. Understanding the lifecycle implications for the development of all aspects of computer-related systems (including software, hardware, and human computer interface).⇒ **Outcome j,m**

C13. Understanding the relationship between quality and lifecycle management⇒ **Outcome c,i,m**

C18. Ability to participate actively and as a member of a team. .⇒ **Outcome d**

C19. Ability to effectively identify the goals and priorities of their project, stating the action, the time and resources required.⇒ **Outcome j**

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome c,i,m**

CS7. Apply the principles of effective information management, information organization, and information-retrieval skills to information of various kinds, including text, images, sound, and video. This must include managing any security issues.⇒ **Outcome d,i,o**

CS9. Identify any risks (and this includes any safety or security aspects) that may be involved in the operation of computing equipment within a given context. ⇒ **Outcome c,d,m**

C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome c**

C11. Understanding of the concept of the lifecycle, including the significance of its phases (planning, development, deployment, and evolution).⇒ **Outcome i,k**

C12. Understanding the lifecycle implications for the development of all aspects of computer-related systems (including software, hardware, and human computer interface).⇒ **Outcome j,m**

C13. Understanding the relationship between quality and lifecycle management⇒ **Outcome c,i,m**

C18. Ability to participate actively and as a member of a team. .⇒ **Outcome d**

C19. Ability to effectively identify the goals and priorities of their project, stating the action, the time and resources required.⇒ **Outcome j**

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome c,i,m**

CS7. Apply the principles of effective information management, information organization, and information-retrieval skills to information of various kinds, including text, images, sound, and video. This must include managing any security issues.⇒ **Outcome d,i,o**

CS9. Identify any risks (and this includes any safety or security aspects) that may be involved in the operation of computing equipment within a given context. ⇒ **Outcome c,d,m**

10. List of topics

1. Software Evolution
2. Software Project Management
3. Software Project Management
4. Software Processes
5. Estándares ISO/IEC

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Software Evolution (12)	
Competences Expected: C7, C11, C12, CS6	
Learning Outcomes	**Topics**
• Identify the principal issues associated with software evolution and explain their impact on the software lifecycle [Familiarity] • Estimate the impact of a change request to an existing product of medium size [Usage] • Use refactoring in the process of modifying a software component [Usage] • Discuss the challenges of evolving systems in a changing environment [Familiarity] • Outline the process of regression testing and its role in release management [Familiarity] • Discuss the advantages and disadvantages of different types of software reuse [Familiarity]	• Software development in the context of large, pre-existing code bases – Software change – Concerns and concernlocation – Refactoring • Software evolution • Characteristics of maintainable software • Reengineering systems • Software reuse – Code segments – Libraries and frameworks – Components – Product lines
Readings : [PM14], [Som10]	

Unit 2: Software Project Management (10)	
Competences Expected: C18, C19, CS7, CS9	
Learning Outcomes	**Topics**
• Discuss common behaviors that contribute to the effective functioning of a team [Familiarity] • Create and follow an agenda for a team meeting [Usage] • Identify and justify necessary roles in a software development team [Usage] • Understand the sources, hazards, and potential benefits of team conflict [Usage] • Apply a conflict resolution strategy in a team setting [Usage] • Use an ad hoc method to estimate software development effort (eg, time) and compare to actual effort required [Usage] • List several examples of software risks [Familiarity] • Describe the impact of risk in a software development lifecycle [Familiarity] • Describe different categories of risk in software systems [Familiarity] • Demonstrate through involvement in a team project the central elements of team building and team management [Usage] • Describe how the choice of process model affects team organizational structures and decision-making processes [Familiarity] • Create a team by identifying appropriate roles and assigning roles to team members [Usage] • Assess and provide feedback to teams and individuals on their performance in a team setting [Usage] • Using a particular software process, describe the aspects of a project that need to be planned and monitored, (eg, estimates of size and effort, a schedule, resource allocation, configuration control, change management, and project risk identification and management) [Familiarity]	• Team participation – Team processes including responsabilities for task, meeting structure, and work schedule – Roles and responsabilities in a software team – Team conflict resolution – Risks associated with virtual teams (communication, perception, structure) • Effort estimation (at the personal level) • Risk – The role of risk in the lifecycle – Risk categories including security, safety, market, financial, technology, people, quality, structure and process • Team management – Team organization and decision-making – Role identification and assigment – Individual and team performance assessment • Project management – Scheduling and tracking – Project management tools – Cost/benefit analysis
Readings : [PM14], [Som10]	

Unit 3: Software Project Management (8)	
Competences Expected: C18, C19, CS7, CS9	
Learning Outcomes	**Topics**
• Track the progress of some stage in a project using appropriate project metrics [Usage] • Compare simple software size and cost estimation techniques [Usage] • Use a project management tool to assist in the assignment and tracking of tasks in a software development project [Usage] • Describe the impact of risk tolerance on the software development process [Assessment] • Identify risks and describe approaches to managing risk (avoidance, acceptance, transference, mitigation), and characterize the strengths and shortcomings of each [Familiarity] • Explain how risk affects decisions in the software development process [Usage] • Identify security risks for a software system [Usage] • Demonstrate a systematic approach to the task of identifying hazards and risks in a particular situation [Usage] • Apply the basic principles of risk management in a variety of simple scenarios including a security situation [Usage] • Conduct a cost/benefit analysis for a risk mitigation approach [Usage] • Identify and analyze some of the risks for an entire system that arise from aspects other than the software [Usage]	• Software measurement and estimation techniques • Software quality assurance and the role of measurements • Risk – Risk identification and management – Risk analysis and evaluation – Risk tolerance (e.g., risk-adverse, risk-neutral, risk-seeking) – Risk planning • System-wide approach to risk including hazards associated with tools
Readings : [PM14], [Som10]	

Unit 4: Software Processes (12)	
Competences Expected: C7, C13, C19, CS6, CS7	
Learning Outcomes	**Topics**
• Describe how software can interact with and participate in various systems including information management, embedded, process control, and communications systems [Usage] • Describe the relative advantages and disadvantages among several major process models (eg, waterfall, iterative, and agile) [Usage] • Describe the different practices that are key components of various process models [Usage] • Differentiate among the phases of software development [Usage] • Describe how programming in the large differs from individual efforts with respect to understanding a large code base, code reading, understanding builds, and understanding context of changes [Usage] • Explain the concept of a software lifecycle and provide an example, illustrating its phases including the deliverables that are produced [Usage] • Compare several common process models with respect to their value for development of particular classes of software systems taking into account issues such as requirement stability, size, and non-functional characteristics [Usage] • Define software quality and describe the role of quality assurance activities in the software process [Usage] • Describe the intent and fundamental similarities among process improvement approaches [Usage] • Compare several process improvement models such as CMM, CMMI, CQI, Plan-Do-Check-Act, or ISO9000 [Usage] • Assess a development effort and recommend potential changes by participating in process improvement (using a model such as PSP) or engaging in a project retrospective [Usage] • Explain the role of process maturity models in process improvement [Usage] • Describe several process metrics for assessing and controlling a project [Usage] • Use project metrics to describe the current state of a project [Usage]	• System level considerations, i.e., the iteraction of software with its intended enviroment • Introduction to software process models (e.g., waterfall, incremental, agile) – Activities with software lifecycles • Programming in the large vs. individual programming • Evaluation of software process models • Software quality concepts • Process improvement • Software process capability maturity models • Software process measurements
Readings : [PM14], [Som10]	

Unit 5: Estándares ISO/IEC (6)	
Competences Expected: C7, C13, C19, CS6, CS7	
Learning Outcomes	**Topics**
• Learn and apply correctly standards and international standards . [Usage]	• ISO 9001:2001. • ISO 9000-3. • ISO/IEC 9126. • ISO/IEC 12207. • ISO/IEC 15939. • ISO/IEC 14598. • ISO/IEC 15504-SPICE. • IT Mark. • SCRUM. • SQuaRE. • CISQ.
Readings : [Som10], [PM14]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** BI0021. Bioinformatics and Biostatistics (Mandatory)
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Alu06] Srinivas Aluru, ed. *Handbook of Computational Molecular Biology.* Computer and Information Science Series. Boca Raton, FL: Chapman & Hall, CRC, 2006.

[CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction.* 279 pages. John Wiley & Sons Ltd., 2000.

[Dur+98] R. Durbin et al. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, 1998, p. 357. ISBN: 9780521629713.

[Kro+94] Anders Krogh et al. "Hidden Markov Models in Computational Biology, Applications to Protein Modeling". In: *J Molecular Biology* 235 (1994), pp. 1501–1531.

[Pev00] Pavel A. Pevzner. *Computational Molecular Biology: an Algorithmic Approach.* Cambridge, Massachusetts: The MIT Press, 2000.

[SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology.* Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296. ISBN: 978-0-534-95262-4.

6. **Information about the course**

(a) **Brief description about the course** The use of computational methods in the biological sciences has become one of the key tools for the field of molecular biology, being a fundamental part of research in this area.
In Molecular Biology, there are several applications that involve both DNA, protein analysis or sequencing of the human genome, which depend on computational methods. Many of these problems are really complex and deal with large data sets.
This course can be used to see concrete use cases of several areas of knowledge of Computer Science such as Programming Languages (PL), Algorithms and Complexity (AL), Probabilities and Statistics, Information Management (IM), Intelligent Systems (IS).

(b) **Prerrequisites:** CS2102. Analysis and Design of Algorithms. (5^{th} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- That the student has a solid knowledge of molecular biological problems that challenge computing.
- That the student is able to abstract the essence of the various biological problems to pose solutions using their knowledge of Computer Science

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

l) Develop principles research in the area of computing with levels of international competitiveness. (**Assessment**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a,b**

C3. An intellectual understanding of, and an appreciation for, the central role of algorithms and data structures.⇒ **Outcome b,l**

C5. Ability to implement algorithms and data structures in software.⇒ **Outcome a,b**

10. List of topics

1. Introduction to Molecular Biology
2. Sequence Comparison
3. Phylogenetic Trees
4. DNA Sequence Assembling
5. Secondary and tertiary structures
6. Probabilistic Models in Molecular Biology

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Introduction to Molecular Biology (4)	
Competences Expected: CS1	
Learning Outcomes	**Topics**
• Achive a general knowledge of the most important topics in Molecular Biology. [Familiarity] • Understand that biological problems are a challenge to the computational world. [Assessment]	• Review of organic chemistry: molecules and macromolecules, sugars, nucleic acids, nucleotides, RNA, DNA, proteins, amino acids and levels of structure in proteins. • The Dogma of Life: From DNA to Proteins, Transcription, Translation, Protein Synthesis. • Genome study: Maps and sequences, specific techniques
Readings : [CB00], [SM97]	

Unit 2: Sequence Comparison (4)	
Competences Expected: CS2	
Learning Outcomes	**Topics**
• Understand and solve the problem of aligning a pair of sequences. [Usage] • Understand and solve the problem of multiple sequence alignment. [Usage] • Know the various algorithms for aligning existing sequences in the literature . [Familiarity]	• Sequences of nucleotides and amino acid sequences. • Sequence alignment, paired alignment problem, exhaustive search, Dynamic programming, global alignment, local alignment, gaps penalty • Comparison of multiple sequences: sum of pairs, complexity analysis by dynamic programming, alignment heuristics, star algorithm, progressive alignment algorithms.
Readings : [CB00], [SM97], [Pev00]	

Unit 3: Phylogenetic Trees (4)	
Competences Expected: CS2	
Learning Outcomes	**Topics**
• Understand the concept of phylogeny, phylogenetic trees and the methodological difference between biology and molecular biology. [Familiarity] • Understand the problem of the reconstruction of phylogenetic trees, to know and apply the main algorithms for the reconstruction of phylogenetic trees. [Assessment]	• Phylogeny: Introduction and phylogenetic relations • Phylogenetic trees: definition, type of trees, problem of search and reconstruction of trees • Reconstruction methods: parsimony methods, distance methods, maximum likelihood methods, confidence of reconstructed trees
Readings : [CB00], [SM97], [Pev00]	

Unit 4: DNA Sequence Assembling (4)	
Competences Expected: CS2	
Learning Outcomes	**Topics**
• Understand the computational challenge of the Sequence Assembly problem. [Familiarity] • Understand the principle of formal model for assembly. [Assessment] • Know the main heuristics for the problem of assembjale of DNA sequences[Usage]	• Biological basis: ideal case, difficulties, alternative methods for DNA sequencing • Formal Assembly Models: Shortest Common Superstring, Reconstruction, Multicontig • Algorithms for sequence assembly: representation of overlaps, paths to create superstrings, voracious algorithm, acyclic graphs. • Assembly heuristics: search for overlays, ordering fragments, alignments and consensus.
Readings : [SM97], [Alu06]	

Unit 5: Secondary and tertiary structures (4)	
Competences Expected: CS2	
Learning Outcomes	**Topics**
• Know the protein structures and the necessity of computational methods for the prediction of the geometry. [Familiarity] • Know the algorithms for solving prediction problems of secondary structures RNA, and structures in proteins. [Assessment]	• Molecular structures: primary, secondary, tertiary, quaternary. • Prediction of secondary structures of RNA: formal model, pair energy, structures with independent bases, solution with Dynamic Programming, structures with loops. • *Protein folding*: Estructuras en proteinas, problema de protein folding. • *Protein Threading*: Definitions, Branch Bound Algorithm, Branch Bound for protein threading. • *Structural Alignment*: Definitions, DALI algorithm
Readings : [SM97], [CB00], [Alu06]	

Unit 6: Probabilistic Models in Molecular Biology (4)	
Competences Expected: CS2	
Learning Outcomes	**Topics**
• Review concepts of Probabilistic Models and understand their importance in Computational Molecular Biology. [Assessment] • Know and apply Hidden Markov Models for various analyzes in Molecular Biology.. [Usage] • Know the application of probabilistic models in Phylogeny and to compare them with non-probabilistic models[Assessment]	• Probability: Random Variables, Markov Chains, Metropoli-Hasting Algorithm, Markov Random Fields, and Gibbs Sampler, Maximum Likelihood. • Hidden Markov Models (HMM), parameter estimation, Viterbi algorithm and Baul-Welch method, Application in paired and multiple alignments, Motifs detection in proteins, in eukaryotic DNA, in sequences families. • Probabilistic phylogeny: probabilistic models of evolution, likelihood of alignments, likelihood for inference, comparison of probailistic and non-probabilistic methods
Readings : [Dur+98], [CB00], [Alu06], [Kro+94]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: ET301. Entrerpreneurship II (Mandatory)
2. Credits: 3
3. Hours of theory and Lab: 2 HT; 2 HP; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[BT06] Leland Blank and Anthony Tarkin. *Ingeniería Económica*. McGraw Hill, México D.F., México, 2006.

[FUP96] Roger Fisher, William Ury, and Bruce Patton. *Si... ¡de acuerdo! Cómo negociar sin ceder*. Norma, Barcelona, 1996.

[KK06] Philip Kotler and Kevin L. Keller. *Dirección de Marketing*. Prentice Hall, México, 2006.

[LW09] Christopher Lovelock and Jochen Wirtz. *Marketing de servicios. Personal, tecnología y estratégia*. Prentice Hall, México, 2009.

[MM06] Fernando de Manuel Dasí and Rafael Martínez-Vilanova Martínez. *Técnicas de Negociación. Un método práctico*. Esic, Madrid, 2006.

6. Information about the course

(a) **Brief description about the course** Este curso tiene como objetivo dotar al futuro profesional de conocimientos, actitudes y aptitudes que le permitan formar su propia empresa de desarrollo de software y/o consultoría en informática. El curso está dividido en tres unidades: Valorización de Proyectos, Marketing de Servicios y Negociaciones. En la primera unidad se busca que el alumno pueda analizar y tomar decisiones en relación a la viabilidad de un proyecto y/o negocio.

En la segunda unidad se busca preparar al alumno para que este pueda llevar a cabo un plan de marketing satisfactorio del bien o servicio que su empresa pueda ofrecer al mercado. La tercera unidad busca desarrollar la capacidad negociadora de los participantes a través del entrenamiento vivencial y práctico y de los conocimientos teóricos que le permitan cerrar contrataciones donde tanto el cliente como el proveedor resulten ganadores. Consideramos estos temas sumamente críticos en las etapas de lanzamiento, consolidación y eventual relanzamiento de una empresa de base tecnológica.

(b) **Prerrequisites:** ET201. Entrerpreneurship I. (8^{th} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- Que el alumno comprenda y aplique la terminología y conceptos fundamentales de ingeniería económica que le permitan valorizar un proyecto para tomar la mejor decisión económica.
- Que el alumno adquiera las bases para formar su propia empresa de base tecnológica.

8. Contribution to Outcomes

d) An ability to function on multidisciplinary teams. (**Usage**)

f) An ability to communicate effectively. (**Usage**)

m) Transform your knowledge of the area of Computer Science into technological enterprises. (**Assessment**)

9. Competences (IEEE)

C17. Ability to properly express in oral and written media as expected from a university graduate. ⇒ **Outcome f**

C18. Ability to participate actively and as a member of a team. .⇒ **Outcome d**

C19. Ability to effectively identify the goals and priorities of their project, stating the action, the time and resources required.⇒ **Outcome m**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome m**

C21. Understanding the professional, legal, security, political, humanistic, environmental, cultural and ethical issues. ⇒ **Outcome m**

C22. Ability to demonstrate attitudes and priorities that honor, protect, and enhance the profession's ethical stature and standing..⇒ **Outcome m**

C23. Ability to undertake, complete, and present a capstone project.⇒ **Outcome m**

C24. Understanding the need for lifelong learning and improving skills and abilities.⇒ **Outcome m**

10. List of topics

1.

2.

3.

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: (20)

Competences Expected: C19

Learning Outcomes	**Topics**
• Permitir al alumno tomar decisiones sobre como invertir mejor los fondos disponibles, fundamentadas en el análisis de los factores tanto económicos como no económicos que determinen la viabilidad de un emprendimiento. [Assessment]	• Introducción • Proceso de toma de decisiones • El valor del dinero en el tiempo • Tasa de interés y tasa de rendimiento • Interés simple e interés compuesto • Identificación de costos • Flujo de Caja Neto • Tasa de Retorno de Inversión (TIR) • Valor Presente Neto (VPN) • Valorización de Proyectos

Readings : [BT06]

Unit 2: (30)

Competences Expected: C20

Learning Outcomes	**Topics**
• Brindar las herramientas al alumno para que pueda identificar, analizar y aprovechar las oportunidades de marketing que generan valor en un emprendimiento. [Usage] • Lograr que el alumno conozca, entienda e identifique criterios, habilidades, métodos y procedimientos que permitan una adecuada formulación de estrategias de marketing en sectores y medios específicos como lo es una empresa de base tecnológica. [Usage]	• Introducción • Importancia del marketing en las empresas de servicios • El Proceso estratégico. • El Plan de Marketing • Marketing estratégico y marketing operativo • Segmentación, targeting y posicionamiento de servicios en mercados competitivos • Ciclo de vida del producto • Aspectos a considerar en la fijación de precios en servicios • El rol de la publicidad, las ventas y otras formas de comunicación • El comportamiento del consumidor en servicios • Fundamentos de marketing de servicios • Creación del modelo de servicio • Gestión de la calidad de servicio

Readings : [KK06], [LW09]

Unit 3: (10)	
Competences Expected: C18	
Learning Outcomes	**Topics**
• Conocer los puntos clave en el proceso de negociación. [Usage] • Establecer una metodología de negociación eficaz. [Usage] • Desarrollar destrezas y habilidades que permitan llevar a cabo una negociación exitosa. [Usage]	• Introducción. ¿Qué es una negociación? • Teoría de las necesidades de la negociación • La proceso de la negociación • Estilos de negociación • Teoría de juegos • El método Harvard de negociación
Readings : [FUP96], [MM06]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS3P02. Cloud Computing (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 1 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[Bal+08] Shumeet Baluja et al. "Video Suggestion and Discovery for Youtube: Taking Random Walks Through the View Graph". In: *Proceedings of the 17th International Conference on World Wide Web*. WWW '08. Beijing, China: ACM, 2008, pp. 895–904. ISBN: 978-1-60558-085-2. DOI: 10.1145/1367497.1367618. URL: http://doi.acm.org/10.1145/1367497.1367618.

[BVS13] Rajkumar Buyya, Christian Vecchiola, and S. Thamarai Selvi. *Mastering Cloud Computing: Foundations and Applications Programming*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013. ISBN: 9780124095397, 9780124114548.

[Cou+11] George Coulouris et al. *Distributed Systems: Concepts and Design*. 5th. USA: Addison-Wesley Publishing Company, 2011. ISBN: 0132143011, 9780132143011.

[HDF11] Kai Hwang, Jack Dongarra, and Geoffrey C. Fox. *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN: 0123858801, 9780123858801.

[Low+12] Yucheng Low et al. "Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud". In: *Proc. VLDB Endow.* 5.8 (Apr. 2012), pp. 716–727. ISSN: 2150-8097. DOI: 10.14778/2212351.2212354. URL: http://dx.doi.org/10.14778/2212351.2212354.

[Mal+10] Grzegorz Malewicz et al. "Pregel: A System for Large-scale Graph Processing". In: *Proc. ACM SIGMOD*. SIGMOD '10 (2010), pp. 135–146. DOI: 10.1145/1807167.1807184. URL: http://doi.acm.org/10.1145/1807167.1807184.

6. **Information about the course**

   (a) **Brief description about the course** In order to understand the advanced computational techniques, the students must have a strong knowledge of the various discrete structures, structures that will be implemented and used in the laboratory in the programming language.

   (b) **Prerrequisites:** CS3700. Big Data. (9th Sem)

   (c) **Type of Course:** Mandatory

   (d) **Modality:** Face to face

7. **Specific goals of the Course**

- That the student is able to model computer science problems using graphs and trees related to data structures.
- That the student apply efficient travel strategies to be able to search data in an optimal way.

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

9. Competences (IEEE)

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a**

C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.⇒ **Outcome b**

C16. Ability to identify advanced computing topics and understanding the frontiers of the discipline.⇒ **Outcome g,i**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome i**

CS3. Analyze the extent to which a computer-based system meets the criteria defined for its current use and future development.⇒ **Outcome j**

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome g,j**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a**

C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.⇒ **Outcome b**

C16. Ability to identify advanced computing topics and understanding the frontiers of the discipline.⇒ **Outcome g,i**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome i**

CS3. Analyze the extent to which a computer-based system meets the criteria defined for its current use and future development.⇒ **Outcome j**

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome g,j**

10. List of topics

1. Distributed Systems
2. Cloud Computing
3. Centros de Procesamiento de Datos

4. Cloud Computing
5. Cloud Computing
6. Modelos de Programación

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Distributed Systems (15)	
Competences Expected: C2, C4	
Learning Outcomes	**Topics**
• Distinguish network faults from other kinds of failures [Familiarity] • Explain why synchronization constructs such as simple locks are not useful in the presence of distributed faults [Familiarity] • Write a program that performs any required marshalling and conversion into message units, such as packets, to communicate interesting data between two hosts [Usage] • Measure the observed throughput and response latency across hosts in a given network [Usage] • Explain why no distributed system can be simultaneously consistent, available, and partition tolerant [Familiarity] • Implement a simple server – for example, a spell checking service [Usage] • Explain the tradeoffs among overhead, scalability, and fault tolerance when choosing a stateful v stateless design for a given service [Familiarity] • Describe the scalability challenges associated with a service growing to accommodate many clients, as well as those associated with a service only transiently having many clients [Familiarity] • Give examples of problems for which consensus algorithms such as leader election are required [Usage]	• Faults (cross-reference OS/Fault Tolerance) – Network-based (including partitions) and node-based failures – Impact on system-wide guarantees (e.g., availability) • Distributed message sending – Data conversion and transmission – Sockets – Message sequencing – Buffering, retrying, and dropping messages • Distributed system design tradeoffs – Latency versus throughput – Consistency, availability, partition tolerance • Distributed service design – Stateful versus stateless protocols and services – Session (connection-based) designs – Reactive (IO-triggered) and multithreaded designs • Core distributed algorithms – Election, discovery
Readings : [Cou+11]	

Unit 2: Cloud Computing (15)

Competences Expected: C2, C4

Learning Outcomes	**Topics**
• Explain the concept of Cloud Computing. [Familiarity] • List some technologies related to Cloud Computing. [Familiarity] • Explain strategies to synchronize a common view of shared data across a collection of devices [Familiarity] • Discuss the advantages and disadvantages of the Cloud Computing paradigm. [Familiarity] • Express the economic benefits as well as the characteristics and risks of the Cloud paradigm for business and cloud providers. [Familiarity] • Differentiate between service models. [Usage]	• Overview of it Cloud Computing. • History. • Overview of the technologies involved. • Benefits, risks and economic aspects. • Cloud services – Infrastructure as a service ∗ Elasticity of resources ∗ Platform APIs – Software as a service – Security – Cost management • Internet-Scale computing – Task partitioning – Data access – Clusters, grids, and meshes

Readings : [HDF11], [BVS13]

Unit 3: Centros de Procesamiento de Datos (10)

Competences Expected: C16

Learning Outcomes	**Topics**
• Describe the evolution of Data Centers. [Familiarity] • Sketch the architecture in detail of the data center. [Familiarity] • Indicate design considerations and discuss their impact. [Familiarity]	• Overview of a data processing center. • Design Considerations. • Comparison of large data processing centers.

Readings : [HDF11], [BVS13]

Unit 4: Cloud Computing (20)	
Competences Expected: CS2, CS3	
Learning Outcomes	**Topics**
• Virtualization – Shared resource management – Migration of processes . [Familiarity] • Explain the advantages and disadvantages of using virtualized infrastructure. [Familiarity] • Identify the reasons why virtualization is becoming enormously useful, especially in the cloud. [Familiarity] • Explain different types of isolation such as failure, resources and security provided by virtualization and used by the cloud. [Familiarity] • Explain the complexity that management can have in terms of abstraction levels and well-defined interfaces and their applicability for virtualization in the cloud. [Familiarity] • Define Virtualization and Identify Different Types of Virtual Machines. [Familiarity] • Identify CPU virtualization conditions, recognize the difference between full virtualization and paravirtualization,explain emulation as a major technique for CPU virtualization and examine virtual CPU planning in Xen. [Familiarity] • Sketching the difference between the classic OS virtual memory and memory virtualization.Explain multiple levels of page mapping as opposed to memory virtualization. Define over-commitment memory and illustrate VMware memory ballooning as a claiming technique for virtualized systems with overcommitted memory. [Familiarity]	• Virtualization – Shared resource management – Migration of processes • Security, resources, and failures isolation . • Storage as a Service. • Elasticity. • Xen y WMware. • Amazon EC2.
Readings : [HDF11], [BVS13]	

Unit 5: Cloud Computing (12)	
Competences Expected: CS2, CS3	
Learning Outcomes	**Topics**
• Describe the general organization of data and storage. [Familiarity] • Identify the problems of scalability and administration of the big data. Discuss several abstractions in storage. [Familiarity] • Compare and contrast different types of file system. Compare and contrast the Hadoop Distributed File System (HDFS) and the Virtual Parallel File System (PVFS). [Usage] • Compare and contrast different types of databases. Discuss the advantages and disadvantages of NoSQL databases. [Usage] • Discuss storage concepts in the cloud. [Familiarity]	• Cloud-based data storage – Shared access to weakly consistent data stores – Data synchronization – Data partitioning – Distributed file systems – Replication • Overview of Storage Technologies. • Fundamentals concepts of cloud storage. • Amazon S3 y EBS. • Distributed File System. • Database System NoSQL.
Readings : [HDF11], [BVS13]	

Unit 6: Modelos de Programación (12)	
Competences Expected: CS6	
Learning Outcomes	**Topics**
• Explain the fundamental aspects of parallel and distributed programming models. [Familiarity] • Differences between programming models: MapReduce, Pregel, GraphLab and Giraph.. [Usage] • Explain the main concepts in the MapReduce programming model. [Usage]	• Overview of cloud computing-based programming models. • Programming Model MapReduce. • Programming model for graph-based applications.
Readings : [HDF11], [BVS13], [Low+12], [Mal+10], [Bal+08]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS3P03. Internet of Things (Mandatory)
2. **Credits:** 3
3. **Hours of theory and Lab:** 1 HT; 4 HL; (15 weeks)
4. **Professor(s)**

Meetings after coordination with the professor

5. **Bibliography**

[KH13] David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach.* 2nd. Morgan Kaufmann, 2013. ISBN: 978-0-12-415992-1.

[Mat14] Norm Matloff. *Programming on Parallel Machines.* University of California, Davis, 2014. URL: `http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf`.

[Pac11] Peter S. Pacheco. *An Introduction to Parallel Programming.* 1st. Morgan Kaufmann, 2011. ISBN: 978-0-12-374260-5.

[Qui03] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP.* 1st. McGraw-Hill Education Group, 2003. ISBN: 0071232656.

[SK10] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming.* 1st. Addison-Wesley Professional, 2010. ISBN: 0131387685, 9780131387683.

6. **Information about the course**

(a) **Brief description about the course** The last decade has an explosive growth in multiprocessor computing, including multi-core processors and distributed data centers. As a result, parallel and distributed computing has evolved from a broadly elective subject to be one of the major components in mesh studies in undergraduate computer science. Both parallel computing and distribution involve the simultaneous execution of multiple processes on different devices that change position.

(b) **Prerrequisites:** CS3P01. Parallel and Distributed Computing . (7^{th} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- That the student is able to create parallel applications of medium complexity by efficiently taking advantage of different mobile devices.

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

9. Competences (IEEE)

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a**

C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.⇒ **Outcome b**

C16. Ability to identify advanced computing topics and understanding the frontiers of the discipline.⇒ **Outcome i**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome i**

CS3. Analyze the extent to which a computer-based system meets the criteria defined for its current use and future development.⇒ **Outcome j**

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome j**

C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome a**

C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.⇒ **Outcome b**

C16. Ability to identify advanced computing topics and understanding the frontiers of the discipline.⇒ **Outcome i**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome i**

CS3. Analyze the extent to which a computer-based system meets the criteria defined for its current use and future development.⇒ **Outcome j**

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome j**

10. List of topics

1. Parallelism Fundamentals
2. Parallel Architecture
3. Parallel Decomposition
4. Communication and Coordination
5. Parallel Algorithms, Analysis, and Programming
6. Parallel Performance

11. Methodology and Evaluation

Methodology:

Theory Sessions:

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
Evaluation System:

12. Content

Unit 1: Parallelism Fundamentals (18)	
Competences Expected: C2	
Learning Outcomes	**Topics**
• Distinguish using computational resources for a faster answer from managing efficient access to a shared resource [Familiarity] • Distinguish multiple sufficient programming constructs for synchronization that may be inter-implementable but have complementary advantages [Familiarity] • Distinguish data races from higher level races [Familiarity]	• Multiple simultaneous computations • Goals of parallelism (e.g., throughput) versus concurrency (e.g., controlling access to shared resources) • Parallelism, communication, and coordination – Parallelism, communication, and coordination – Need for synchronization • Programming errors not found in sequential programming – Data races (simultaneous read/write or write/write of shared state) – Higher-level races (interleavings violating program intention, undesired non-determinism) – Lack of liveness/progress (deadlock, starvation)
Readings : [Pac11], [Mat14], [Qui03]	

Unit 2: Parallel Architecture (12)	
Competences Expected: C4	
Learning Outcomes	**Topics**
• Explain the differences between shared and distributed memory [Assessment] • Describe the SMP architecture and note its key features [Assessment] • Characterize the kinds of tasks that are a natural match for SIMD machines [Usage] • Describe the advantages and limitations of GPUs vs CPUs [Usage] • Explain the features of each classification in Flynn's taxonomy [Usage] • Describe the challenges in maintaining cache coherence [Familiarity] • Describe the key performance challenges in different memory and distributed system topologies [Familiarity]	• Multicore processors • Shared vs distributed memory • Symmetric multiprocessing (SMP) • SIMD, vector processing • GPU, co-processing • Flynn's taxonomy • Instruction level support for parallel programming – Atomic instructions such as Compare and Set • Memory issues – Multiprocessor caches and cache coherence – Non-uniform memory access (NUMA) • Topologies – Interconnects – Clusters – Resource sharing (e.g., buses and interconnects)
Readings : [Pac11], [KH13], [SK10]	

Unit 3: Parallel Decomposition (18)	
Competences Expected: C16	
Learning Outcomes	**Topics**
• Explain why synchronization is necessary in a specific parallel program [Usage] • Identify opportunities to partition a serial program into independent parallel modules [Familiarity] • Write a correct and scalable parallel algorithm [Usage] • Parallelize an algorithm by applying task-based decomposition [Usage] • Parallelize an algorithm by applying data-parallel decomposition [Usage] • Write a program using actors and/or reactive processes [Usage]	• Need for communication and coordination/synchronization • Independence and partitioning • Basic knowledge of parallel decomposition concept • Task-based decomposition – Implementation strategies such as threads • Data-parallel decomposition – Strategies such as SIMD and MapReduce • Actors and reactive processes (e.g., request handlers)
Readings : [Pac11], [Mat14], [Qui03]	

Unit 4: Communication and Coordination (18)	
Competences Expected: C16	
Learning Outcomes	**Topics**
• Use mutual exclusion to avoid a given race condition [Usage] • Give an example of an ordering of accesses among concurrent activities (eg, program with a data race) that is not sequentially consistent [Familiarity] • Give an example of a scenario in which blocking message sends can deadlock [Usage] • Explain when and why multicast or event-based messaging can be preferable to alternatives [Familiarity] • Write a program that correctly terminates when all of a set of concurrent tasks have completed [Usage] • Give an example of a scenario in which an attempted optimistic update may never complete [Familiarity] • Use semaphores or condition variables to block threads until a necessary precondition holds [Usage]	• Shared Memory • Consistency, and its role in programming language guarantees for data-race-free programs • Message passing – Point-to-point versus multicast (or event-based) messages – Blocking versus non-blocking styles for sending and receiving messages – Message buffering (cross-reference PF/Fundamental Data Structures/Queues) • Atomicity – Specifying and testing atomicity and safety requirements – Granularity of atomic accesses and updates, and the use of constructs such as critical sections or transactions to describe them – Mutual Exclusion using locks, semaphores, monitors, or related constructs ∗ Potential for liveness failures and deadlock (causes, conditions, prevention) – Composition ∗ Composing larger granularity atomic actions using synchronization ∗ Transactions, including optimistic and conservative approaches • Consensus – (Cyclic) barriers, counters, or related constructs • Conditional actions – Conditional waiting (e.g., using condition variables)
Readings : [Pac11], [Mat14], [Qui03]	

Unit 5: Parallel Algorithms, Analysis, and Programming (18)	
Competences Expected: CS2	
Learning Outcomes	**Topics**
• Define “critical path”, “work”, and “span” [Familiarity] • Compute the work and span, and determine the critical path with respect to a parallel execution diagram [Usage] • Define “speed-up” and explain the notion of an algorithm’s scalability in this regard [Familiarity] • Identify independent tasks in a program that may be parallelized [Usage] • Characterize features of a workload that allow or prevent it from being naturally parallelized [Familiarity] • Implement a parallel divide-and-conquer (and/or graph algorithm) and empirically measure its performance relative to its sequential analog [Usage] • Decompose a problem (eg, counting the number of occurrences of some word in a document) via map and reduce operations [Usage] • Provide an example of a problem that fits the producer-consumer paradigm [Usage] • Give examples of problems where pipelining would be an effective means of parallelization [Usage] • Implement a parallel matrix algorithm [Usage] • Identify issues that arise in producer-consumer algorithms and mechanisms that may be used for addressing them [Usage]	• Critical paths, work and span, and the relation to Amdahl’s law • Speed-up and scalability • Naturally (embarrassingly) parallel algorithms • Parallel algorithmic patterns (divide-and-conquer, map and reduce, master-workers, others) – Specific algorithms (e.g., parallel MergeSort) • Parallel graph algorithms (e.g., parallel shortest path, parallel spanning tree) (cross-reference AL/Algorithmic Strategies/Divide-and-conquer) • Parallel matrix computations • Producer-consumer and pipelined algorithms • Examples of non-scalable parallel algorithms
Readings : [Mat14], [Qui03]	

Unit 6: Parallel Performance (18)	
Competences Expected: CS3	
Learning Outcomes	**Topics**
• Detect and correct a load imbalance [Usage] • Calculate the implications of Amdahl's law for a particular parallel algorithm (cross-reference SF/Evaluation for Amdahl's Law) [Usage] • Describe how data distribution/layout can affect an algorithm's communication costs [Familiarity] • Detect and correct an instance of false sharing [Usage] • Explain the impact of scheduling on parallel performance [Familiarity] • Explain performance impacts of data locality [Familiarity] • Explain the impact and trade-off related to power usage on parallel performance [Familiarity]	• Load balancing • Performance measurement • Scheduling and contention (cross-reference OS/Scheduling and Dispatch) • Evaluating communication overhead • Data management – Non-uniform communication costs due to proximity (cross-reference SF/Proximity) – Cache effects (e.g., false sharing) – Maintaining spatial locality • Power usage and management
Readings : [Pac11], [Mat14], [KH13], [SK10]	

Universidad Nacional Mayor de San Marcos
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. Code and Name: CS4004. Final Project III (Mandatory)
2. Credits: 6
3. Hours of theory and Lab: 2 HT; 8 HP; (15 weeks)
4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[Ass08] Association for Computing Machinery. *Digital Libray*. http://portal.acm.org/dl.cfm. Association for Computing Machinery, 2008.

[Cit08] CiteSeer.IST. *Scientific Literature Digital Libray*. http://citeseer.ist.psu.edu. College of Information Sciences and Technology, Penn State University, 2008.

[IEE08] IEEE-Computer Society. *Digital Libray*. http://www.computer.org/publications/dlib. IEEE-Computer Society, 2008.

6. Information about the course

(a) **Brief description about the course** This course aims to enable students to complete properly their draft of thesis.

(b) **Prerrequisites:** CS4003. Final Project II. (9^{th} Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. Specific goals of the Course

- That the student completes this course with his thesis elaborated in sufficient quality as for an immediate support.
- That the student formally present the draft dissertation before the authorities of the faculty
- The deliverables of this course are:

  Parcial: Advancement of the thesis project including in the document: introduction, theoretical framework, state of the art, proposal, analysis and / or experiments and solid bibliography.

  Final: Full thesis document and ready to support in a period of no more than fifteen days.

8. Contribution to Outcomes

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Assessment**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Assessment**)

f) An ability to communicate effectively. (**Assessment**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Assessment**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

l) Develop principles research in the area of computing with levels of international competitiveness. (**Assessment**)

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Assessment**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Assessment**)

f) An ability to communicate effectively. (**Assessment**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Assessment**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

l) Develop principles research in the area of computing with levels of international competitiveness. (**Assessment**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a,b,c**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome e,f.g**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome h,i,l**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a,b,c**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome e,f.g**

CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.⇒ **Outcome h,i,l**

10. List of topics

1. Escritura del Borrador del trabajo de final de carrera (tesis)

11. Methodology and Evaluation

Methodology:

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Escritura del Borrador del trabajo de final de carrera (tesis) (60)	
Competences Expected: C1,C20,CS2	
Learning Outcomes	**Topics**
• Experimental part completed (if appropriate to the project) [Assessment] • Verify that the document complies with the thesis format of the course [Assessment] • Delivery of the completed thesis draft and considered ready for public support (approval requirement)[Assessment]	• Writing and correction of the work of end of career
Readings : [IEE08], [Ass08], [Cit08]	