ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Faculty of Electrical and Computer Engineering
COURSE SYLLABUS
Algorithm Analysis

1. CODE AND NUMBER OF CREDITS

CODE	FIEC04366	
NUMBER OF CREDITS: 4	**Theoretical: 4**	**Practical: 0**

2. COURSE DESCRIPTION

This is an introductory course to the analysis and design of algorithms, which will explore methods to design efficient solutions to important problems in computer science; it will also be studied methods to analyze these solutions and the mathematical tools required. Given a problem several solutions will be proposed and analyzed, later implemented for the purpose of verifying their performances and then select the "best". This course will emphasize the importance of calculating the resources necessary to run an algorithm, and of proving that it is a correct. Finally, and from a formal point of view, it will explore concepts such as: problem, algorithm, an algorithm efficiency, classification of problems, etc.

3. PRE-REQUISITES AND CO-REQUISITES

PRE-REQUISITES	FIEC03012 DATA STRUCTURES ICM00901 DISCRETE MATHEMATICS
CO-REQUISITES	

4. CORE TEXT AND OTHER REQUIRED REFERENCES FOR THE TEACHING OF THE COURSE

CORE TEXT	1. Corman Y. H., Leiserson C. E., Rivest R. L. y Stein C. Introduction to Algorithms, Third Edition, MIT-Mc Graw Hill, 2009
REFERENCES	1. Algorithms, Brassard G. and Bratley P., Prentice Hall Internacional. 1988 2. Computer Algorithms: Introduction to Design and Analyisis, Sara Baase and Allen Van Gelder. Addison Wesley, 199 3. Introduction to the Design and Analysis of Algorithms. Anany Levitin. Addison Wesley, 2002

5. COURSE LEARNING OUTCOMES

At the end of the course, the student will be able to:

1. Apply mathematical concepts to analyze an algorithm: proofs, asymptotic notations, recurrence equations, and others that are required.
2. Explain why it is necessary to prove that an algorithm is correct.
3. Calculate the resources necessary to run an algorithm: memory and running time.
4. Identify, explain and apply the following design methods: Divide and Cconquer, Dynamic Programming and Greedy Algorithms.
5. Implement and evaluate different algorithmic solutions for the same problem in order to decide which is most suitable.
6. Design, implement and evaluate solutions to some important problems in computer science: sort, search, calculate minimum distances in a graph, find a least-cost spanning tree, find a code of optimal average length, knapsack problem, calculate the distance between two words, etc.
7. Define, explain and illustrate the classes of problems: P, NP, NPC.

6. COURSE PROGRAM

I. Introduction (2 sessions - 4 hours).
- Introduction to the course Introduction to the analysis of algorithms

II. Asymptotic notations (2 sessions - 4 hours).
- The notations great or, great Omega and theta notations or small, small omega

III. Recurrence equations (2 sessions - 4 hours).
- Method substitution method by iteration
- method of the theorem master

IV. Divide and conquer (2 sessions - 4 hours).
- Description of the method
- Sort by Mergesort
- Analysis of Mergesort,

V. Quicksort (2 sessions - 4 hours).
- Description of quicksort
- Analysis of the best and the worst cases
- Average time of execution

VI. Rank and Selection (2 sessions - 4 hours).
- Selection Algorithm

VII. Linear sort (2 sessions - 4 hours).
- Counting sort
- Bucket sort
- Radix sort

VIII. Disjoint Sets (2 sessions - 4 hours).
- Data Structures and Operators

IX. Dynamic Programming (2 sessions - 4 hours).
- Description of the Method
- The Floyd -Warshall Algorithm
- The problem of chain matrix multiplication
- The problem of finding the common substring of maximum length
- The knapsack problem

X. Greedy Algorithms (2 sessions - 4 hours).
- Description of the Method
- The problem of finding a minimum cost spanning tree
- The Huffman Algorithm

XI. Amortization Analysis (3 sessions - 6 hours).
- The aggregate method
- The method of the potential

XII. Tree structures and algorithms (3 sessions - 6 hours).
- Binary trees
- Red-black trees.

XIII. Computational complexity analysis (2 sessions - 4 hours).
- Problems, algorithms, Turing machines
- P,NP, NPC problems

7. WORKLOAD: THEORY/PRACTICE

2 sessions of 2 hours each week. Tuesdays and Thursdays, 9:30-11:30

8. CONTRIBUTION OF THE COURSE TO THE EDUCATION OF THE STUDENT

We review some math concepts useful to analyze algorithms: demonstrations, recurrence equations, asymptotic notations, trees, graphs, automata.

We review some basic computer science concepts, such as: data structures, algorithms, programming, computability, Turing machines.

Students analyze and implement several algorithms that solve the same problem, in order to compare their performances and choose the best.

Students design efficient solutions to important problems in the field of computer science.

Students implement their designs in some high-level programming language: C, C++, Java, Scala or Python.

BASIC TRAINING	PROFESSIONAL TRAINING	SOCIAL SKILLS DEVELOPMENT
	X	

9. THE RELATIONSHIP BETWEEN THE LEARNING OUTCOMES OF THE COURSE AND THE LEARNING OUTCOMES OF THE DEGREE PROGRAM

LEARNING OUTCOMES OF THE DEGREE PROGRAM*	CONTRIBUTION (High, Medium, Low)	LEARNING OUTCOMES OF THE COURSE**	THE STUDENT MUST:
a) An ability to apply knowledge of mathematics, science and engineering.	High	1,2,3,4,5,6,7	Prove that a function f (n) belongs to some asymptotic notation. Solve recurrence equations to calculate the time of the execution of an algorithm. Explain the importance of proving that an algorithm is correct. Compute the memory required to the run an algorithm. Calculate algorithms running times: best and worst case, average. Explain and apply the following design methods: Divide and Conquer, Dynamic Programming and Greedy Algorithms. Design, implement and evaluate solutions to important problems in computer science, for example: sort, search, minimum distances in a graph, find a minimum-cost spanning tree, Huffman codes, knapsack problem, minimum distance between two words, etc. Define, explain and illustrate with examples the classes of problems: P, NP, NPC. Explain the importance of recognizing that a problem is NPC.
b) An ability to design and conduct experiments, and to analyze and interpret data	---		
c) An ability to design a system, component or process to satisfy realistic constraints.	---		
d) An ability to function on multidisciplinary teams.	---		
e) An ability to identify, formulate and solve engineering problems.	---		
f) An understanding of ethical and professional responsibility.	---		
g) An ability to communicate effectively.	---		
h) A broad education necessary to understand the impact of engineering solutions in a social, environmental, economic and global context.	---		
i) A recognition of the need for, and an ability to engage	---		

in life-long learning.			
j) A knowledge of contemporary issues.	---		
k) An ability to use the techniques, skills, and modern tools necessary for engineering practice.	---		
l) Capacity to lead, manage and undertake projects.	---		

10. EVALUATION IN THE COURSE

Evaluation activities	
Exams	X
Tests	X
Homework/tasks	X
Projects	X
Laboratory/Experiments	
Class participation	X
Visits	
Other	

11. PERSON RESPONSIBLE FOR THE CREATION OF THE SYLLABUS AND THE DATE OF ITS CREATION

Created by	Carlos Jordán
Date	August, 2012

12. APPROVAL

ACADEMIC SECRETARY OF THE ACADEMIC DEPARTMENT	DIRECTOR OF TECHNICAL ACADEMIC SECRETARY
NAME: Sra. Leonor Caicedo G.	NAME: Ing. Marcos Mendoza V.
SIGNATURE:	SIGNATURE: ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL *Ing. Marcos Mendoza V.* DIRECTOR DE LA SECRETARIA TÉCNICA ACADÉMICA
Date of approval by the Directive Council: 2013-334 2013-08-12	

13. VALIDITY OF THE SYLLABUS

RESOLUTION OF THE POLYTECHNIC BOARD:	13-10-269
DATE:	2013-10-17