

Academic Year: (2020 / 2021)

Review date: 10/07/2020 21:54:18

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: ALONSO WEBER, JUAN MANUEL

Type: ECTS Credits : 6.0

Year : 4 Semester :

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Automata and Formal Language Theory
Discrete Mathematics

OBJECTIVES

The aim of this course is that students acquire the following skills:

1. Transverse (Generic) competences

- Ability to analyze and synthesize.
PO: a,e,g
- Problem Solving
PO: a,e
- Critical Reasoning.
PO: b,g
- Teamwork
PO: b,e,g
- Written communication.
PO: g
- CB4 Students can communicate information, ideas, problems and solutions to both specialist and non-specialist audiences.

2. Specific competences for Learning**a.- Cognitive (To Know).**

- PO: a
- To Understand the formal definition of computation.
 - To understand the limits of computation
 - To know the methods to calculate the computational complexity.
 - To know what could be efficiently computed
 - To Understand what reducing one problem to another is.

b.- Procedural

- PO: a,b,e,g,h,i
- To fix what can be efficiently computed
 - To fix if a problem can be efficiently computed
 - To fix which algorithm is the most efficient
 - To select the most efficient data structure and the most suitable programming technique for designing an efficient algorithm
 - To fix the simplest model of computation for a problem
 - Ability to transform a formal statement into a informal statement and vice versa

- CB5 Students have developed those learning skills needed

to undertake a further study with a high degree of autonomy.

- CECRI6 Students must learn about basic algorithmic procedures, and its application, for computing technologies to design solutions to problems, analyzing the appropriateness and complexity of the proposed algorithms.

- CECC1 Students must achieve the knowledge of the fundamental principles and models of computers, and know how to apply them to interpret, select, estimate, model, and create new concepts, theories, uses and developments computer-related technology.

- CECC3 Students must achieve the ability to evaluate the computational complexity of a problem, to know algorithmic strategies that can lead to its resolution and to recommend, develop and implement them to ensure that the best performance according to the requirements.

c.- Attitude

PO: a,b,e,g,h,i

- Ability to analyze computational problems and its solutions.
- Concern on quality
- Interest for finding new alternatives

DESCRIPTION OF CONTENTS: PROGRAMME

1. Cost of Algorithms.
 - 1.1 Computational Complexity.
 - 1.2 Computational Cost: Structured Programming
 - 1.3 Recursive Algorithms
 - 1.4 Probabilistic Analysis
2. Introduction to Computability Theory
 - 2.1 Problem: definition
 - 2.2 Turing Machine: decidability
 - 2.3 Class of Problems (Computability)
3. Introduction to Complexity Theory
 - 3.1 Problem Reducction
 - 3.2 Classes P, NP y NP-Complete.
 - 3.3 Classes PSpace, NPSpace.
 - 3.3 Classes NP-Hard,Exp,CoP, CoNP.
4. Other Models of Computation
 - 4.1 Lambda-Calculus
 - 4.2 RAM Model
 - 4.2 Cellular Automata
 - 4.3 Lindenmayer Systems
 - 4.4 Randomized Algorithms

LEARNING ACTIVITIES AND METHODOLOGY

Theoretical lectures: 1.5 ECTS.

These classes constitute a guide for students to achieve cognitive skills, and acquire the basic elements to develop procedural skills. A part of the ECTS corresponds to the load of autonomous work carried out by the students.

PO: a,b,e,h,i

Practical Lectures and Labs: 1.5 ECTS

They contribute to the development of the generic competences and the application of the attitude ones. These classes consist on developing and solving practical cases (exercises, problems and practices) which are aimed at achieving the procedural competences. An important part of the ECTS corresponds to the autonomous work of the student.

PO: a,b,e,g,h,i

Realización de Actividades Académicas Other academic activities

Present teacher: 1.0 ECTS.

Resolution of small questions and exercises.

Practical Work

PO: a,e,g,

Absent teacher: 1.5 ECTS.

Readings about the field as well as exercises, problems and practices related to the theoretical lectures and practical classes.

PO: a,b,e,g,h,i

Exam: 0.5 ECTS.

Preparation and realization of the examination, in which the level achieved by the student in relation to the established competences is assessed.

PO: a,b,e,g

ASSESSMENT SYSTEM

% end-of-term-examination/test:	35
--	----

% of continuous assessment (assignments, laboratory, practicals...):	65
---	----

The evaluation will consist in continuous assessment and a final practice or a final exam.

The continuous evaluation will address:

- practical work done by the students.

PO: a, b, e, g, h, i

- academic activities in the presence of the professor.

PO: a, e, g

- several partial exams to evaluate the students' work individually

For those students that pass the Continuous Evaluation, it will be possible carry out a final project (35% of the total grade mark).

Otherwise, the students must be examined in a final-term exam

A Final-term exam to assess the extent to which the students have acquired the cognitive and procedural competences

It will consist of theoretical questions, exercises and practical problems

PO: a,b,e,gks

A minimum grade for the final practice or final-term exam (4 out of 10) is required to pass the subject.

In case it is not possible to carry out final-term exams with the required guarantees of integrity, there is the possibility of replacing them with a final project.

This set of measures considers the foreseeable cases in the context imposed by the covid-19.

BASIC BIBLIOGRAPHY

- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. Introduction to Automata Theory, Languages, and Computation, Addison-Wesley 3rd Edition.

- Michael Sipser. Introduction to the Theory of Computation. 2nd ed., Boston, MA: Course Technology, 2005. ISBN: 0534950973..

- S. Wolfram. Cellular Automata and Complexity., Addison-Wesley, (1996).

ADDITIONAL BIBLIOGRAPHY

- C. Papadimitriou Computational Complexity., Addison-Wesley, 1995.

- H. S. Wilf Algorithms and Complexity., Prentice-Hall, 1986.

- Jeffrey Shallit. A Second Course in Formal Languages and Automata Theory., Cambridge University Press..