Programming

Academic Year: (2023 / 2024) **Review date: 07/06/2023 14:09:26**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: TOLEDO HERAS, MARIA PAULA DE

Type: Basic Core ECTS Credits : 6.0

Year : 1 Semester : 1

Branch of knowledge: Engineering and Architecture

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

(None)

LEARNING OUTCOMES

CB1. Students have demonstrated possession and understanding of knowledge in an area of study that builds on the foundation of general secondary education, and is usually at a level that, while relying on advanced textbooks, also includes some aspects that involve knowledge from the cutting edge of their field of study.
CB2. Students are able to apply their knowledge to their work or vocation in a professional manner and possess the competences usually demonstrated through the development and defence of arguments and problem solving within their field of study.
CB3. Students have the ability to gather and interpret relevant data (usually within their field of study) in order to make judgements which include reflection on relevant social, scientific or ethical issues.
CB4. Students should be able to communicate information, ideas, problems and solutions to both specialist and non-specialist audiences.
CB5. Students will have developed the learning skills necessary to undertake further study with a high degree of autonomy.
CG10. Being able to work in a multi-lingual and multidisciplinary environment
CE3 Módulo FB. Basic knowledge of the use and programming of computers, operating systems, databases, and computer software with engineering applications.
CT1. Ability to communicate knowledge orally as well as in writing to a specialized and non-specialized public.
CT2. Ability to establish good interpersonal communication and to work in multidisciplinary and international teams.
CT3. Ability to organize and plan work, making appropriate decisions based on available information, gathering and interpreting relevant data to make sound judgement within the study area.
CT4. Motivation and ability to commit to lifelong autonomous learning to enable graduates to adapt to any new situation.

By the end of this content area, students will be able to have:
RA1.1 knowledge and understanding of the programming foundations and computer systems underlying their branch of engineering;
RA1.4 awareness of the wider multidisciplinary context of engineering.
RA2.1 the ability to apply their knowledge and understanding to identify, formulate and solve engineering problems using computer-aided methods;
RA5.2 the ability to combine theory and practice to solve engineering problems using computer-aided methods.

OBJECTIVES

- Synthesis and analysis
- Organization and planning
- Application of theoretical knowledge to practical problems
- Use of computers
- Proactivity, creativity and reasoning for problem resolution

- Team work

- Cognitive:
  - o To understand the role of Computer Science and Programming in the context of Industrial Engineering
  - o To solve engineering problems by designing algorithms and developing computer programs
  - o To understand the fundamentals of structured and modular programming
  - o To apply theoretical knowledge to solve practical problems by implementing computer programs in the C programming language
  - o To understand the role of Computer Science and Programming in the context of Industrial Engineering

- Procedural:
  - o To use a personal computer
  - o To solve engineering problems by designing and implementing computer programs in the C language
  - o To acquire a good programming style, resulting in efficient, well-organized, and well-documented programs
  - o To acquire the ability to understand and use third-party programs
  - o To learn how to use an integrated development environment (IDE) for program coding, compiling, and debugging
  - o To learn how to use common C programming libraries.

DESCRIPTION OF CONTENTS: PROGRAMME

Topic 1. Introduction to computer science and programming.

- Computer science and computers. A historical perspective.
- Information representation in computers
- Algorithms and programs. Tools for algorithm design

Topic 2. Software and Hardware

- Programs and programming languages. Classification. Translators, compilers and interpreters. Operating systems.
- Functional structure of a computer. Internal architecture. Instruction execution by the processor. Data storage. Peripherals. Computer networks and the Internet

Topic 3. Basic elements of the C programming language.

- General structure of a program
- Variables and constants
- Types of operators: arithmetic, relational, logic and assignment operators.
- Operators, expressions and instructions
- Pointer type.
- Input and output instructions

Topic 4. Control Flow and Loops

- Selection structures: if-else, switch
- Repetition structures (loops): for, while, do-while
- Nested control structures

Topic 5. Functions

- Modular programming
- Function definition
- Calling a function
- Types of arguments: input, output, input / output
- Passing Arguments by Value or by Reference -Scope of Function Variables. Visibility
- Arrays and structures as parameters
- Library functions and standard C libraries

Topic 6. Complex Data Types

- Introduction: structured vs simple data types
- Definition and use of arrays
- Pointers and arrays
- Character strings
- User defined data structures: records
- Arrays of records

Topic 7. Search, sort and merge algorithms

- Search algorithms
- Sort algorithms
- Merge algorithms

Topic 8. Advanced Topics

- External data structures: files and databases
- Dynamic memory allocation
- Computer programs commonly used in engineering.

LEARNING ACTIVITIES AND METHODOLOGY

Theory classes:

Basic theoretical knowledge and skills will be presented in large groups.

Resolution of Exercises:

Resolution of exercises by the student that will serve as self-evaluation and to acquire the procedural and cognitive competences.

Laboratory sessions:

Small groups classes, in which problems proposed to the students are discussed and developed using the computer.

Tutorials

The student will ask for a tutorial always that it is needed.

Final Project.

Final project in teams, to develop a longer and more complex program

ASSESSMENT SYSTEM

% end-of-term-examination/test:	50
% of continuous assessment (assigments, laboratory, practicals…):	50

Continuous Assessment Exams:

50% of the course grade is assigned by continous assessment
Continuous assessment will consist of written exams or team projects or both

Final exam:

Value: 50% of the final grade

The final exam mark must be higher than 4.0 to pass the course.

In case the exam grade is lower than 4 but the continuous assessment is passed and the average is higher than 4, the student will be graded with "Fail - 4.5".

% end-of-term-examination/test:	50
% of continuous assessment (assigments, laboratory, practicals…):	50

Second exam call (convocatoria extraordinaria):

In the second exam session, an exam will be held with the same structure as in the ordinary exam session. In this exam session, the mark of the continuous assessment will only be taken into account if it improves the mark of the extraordinary exam.

BASIC BIBLIOGRAPHY

- Al Kelley and Ira Pohl A book on C : programming in C , Addison-Wesley, 1998
- Brian W. Kernighan, Dennis M. Ritchie C Programming Language, Prentice Hall, 1988
- Deitel, Harvey M. C : how to program, Prentice-Hall International, 1994
- Greg Perry, Dean Miller C Programming Absolute Beginner´s Guide., Que, 2013
- K. N. King C Programming: A Modern Approach, W.W. Norton & Company, 2008
- King, K.K. C programming: a modern approach, W.W. Norton & Company, 2008
- Paul J. Deitel, Harvey M. Deitel C: How to Program, Prentice Hall, 2009

ADDITIONAL BIBLIOGRAPHY

- Doris Appleby, Julius J. Vandekopple Lenguajes de Programación: Paradigma y práctica, McGraw-Hill, 1998
- Yung-Hsiang Lu Intermediate C Programming, CRC Press, 2015