Programming

Academic Year: (2021 / 2022) **Review date: 04-06-2021**

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)

-

OBJECTIVES

General Description

This course provides an overview of fundamental elements in computer Programming. Students will learn the basis programming, analysis of problems and algorithm development.

The use of Matlab in the lab sessions will provide to the students a high-level and interactive integrated development environment plus a a thecnical computing language for algorithm development, and its implementation and debugging in source code in Matlab programming language.

General Competences

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

Specific Competences and Learning Objectives
---

* Cognitive:

1. Identify and explain the fundamental topics in the field of Computer Programming, Structured Programming, that facilitate the capabilities to learn new methods and theories in the field of Aerospace Engineering.

* Procedural:
- Design Algorithms

- To learn how to use an integrated development environment (IDE) for program coding, and debugging the algorithm and the source code

- To acquire a good programming style, resulting in efficient, well-organized, and well-documented programs

- To acquire the ability to understand and use third-party programs

- Use the knowledge and abilities gained by personal exertion about computer programming

(i.e. algorithm development) to solve problems, and to adapt to the changes that technology development will bring, in the field of Aeronautical Engineering.

DESCRIPTION OF CONTENTS: PROGRAMME

UNIT 1 INTRODUCTION TO PROGRAMMING
1.- Binary Code
2.- Basic elements of programming
  2.1.- Base instruction
  2.2.- Scalar data type
  2.3.- Arithmetic and logic expressions
3.- Structured Programming.
  3.1.- Theorem of Structured Programming
  3.2.- Flow Charts.
4.- Conditional Intructions
  4.1.- IF instruction
  4.1.- Switch instruction
5.- Loops
  5.1.- For
  5.2.- While
  5.3.- Nested Loops
6.- Debugging
7.- Arrays
  7.1 Vectors (one-dimensional array)
  7.2 Matrices and multi-dimensional arrays
8.- Functions
  8.1.- Definition and function call
  8.2.- Recursion
9.- Structures
  9.1 Definition.
  9.2 Arrays of structures
10.- Search, Sorting and Merge
  10.1 Linear and Binary Search
  10.2 Sorting Algorithms
  10.3 Merge
11.- Files
  11.1 Binary and plain text files
  11.2 Reading and writing files.

UNIT 2 INTRODUCTION TO COMPUTER SCIENCE
1.- Information Technology
2.- Software
3.- Data Structures (databases)
4.- Operating Systems
5.- Hardware
6.- Telecommunications

LEARNING ACTIVITIES AND METHODOLOGY

The learning activities in lectures and labs sessions are organized as follows:
- lectures (50% of classroom learning activities): theoretical aspects, and exercises
- lab sessions (50% of classroom learning activities): exercises and problems
- personal work, including solving problems on the computer and in writing

The student must bear in mind that his/her personal work outside the classroom (study the theoretical concepts, exercises, and problems) is a keylearning activity, and it makes up for around 50% of the work in this course

ASSESSMENT SYSTEM

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

The course follows a continuous assessment system:
- Continuous assessment = Tests and problems (at least two mid-term exams): 50%
- End of term exam: 50%

If the mark in the final exam is below 4.0 the student doesn't pass the course and the final mark will be no higher than 4.5.

By University policy, students can re-sit the exam at the end of the academic year (convocatoria extraordinaria), being the final grade the mark in this exam.
However, the continous assessment mark can be considered if the exam is passed (mark higher that 5)
the mark including the continuous assessment is higher than the mark in the exam, the best case for the student will prevail
To summarize:
- The exam (re-sit) has a weight of 100%
- If the student followed the continuous assessment, and the mark in the exam is 5 or higher, the final mark will be the best option for the student between the exam (re-sit) mark and the mark including the continuous assessment (50% exam, 50% continuous assessment)

BASIC BIBLIOGRAPHY

- Karl Beecher Computational Thinking - A beginner's guide to problem-solving and programming, BCS Learning & Development Limited.

- Ravi Sethi. Programming languages, concepts and constructs. , Addison-Wesley.

- Stormy Attaway Matlab: A Practical Introduction to Programming and Problem Solving, 2nd Edition, Butterworth-Heinemann.

ADDITIONAL BIBLIOGRAPHY

- Behrouz Forouzan and Firouz Mosharraf Foundations of Computer Science , Cengage.

- ITL Education Solutions Limited Introduction to Information Technology , Pearson Education India.

- V. Rajaraman Introduction To Information Technology, Prentice-Hall of India Pvt.Ltd.

BASIC ELECTRONIC RESOURCES

- Karl Beecher . Computational Thinking - A beginner's guide to problem-solving and programming BCS Learning & Development Limited: https://proquest.safaribooksonline.com/book/programming/9781780173641?bookview=overview

- Mathworks . Mathworks educación: https://es.mathworks.com/academia.html?s_tid=gn_acad

- Stormy Attaway . Matlab: A Practical Introduction to Programming and Problem Solving, 2nd Edition utterworth-Heinemann: https://proquest.safaribooksonline.com/book/computer-aided-engineering/9780123850812