

Programming

Academic Year: (2019 / 2020)

Review date: 15-10-2018

Department assigned to the subject: Department of Computer Science and Engineering

Coordinating teacher: IGLESIAS MAQUEDA, ANA MARIA

Type: Basic Core ECTS Credits : 6.0

Year : 1 Semester : 1

Branch of knowledge: Engineering and Architecture

STUDENTS ARE EXPECTED TO HAVE COMPLETED

No pre-requisites

COMPETENCES AND SKILLS THAT WILL BE ACQUIRED AND LEARNING RESULTS.

1. Basics knowledge on computer systems and industrial software applications.
2. Defining basic algorithms to fulfil a set of requirements.
3. Constructing small applications by using a specific programming language.
4. Prepare, write and execute testing cases.

DESCRIPTION OF CONTENTS: PROGRAMME

DESCRIPTION OF CONTENTS:

The purpose of the course is to give students an overview on programming techniques. As programming language, it will be used a imperative programming language. The language used during this academic course is Python

PROGRAMME:

1. Programming foundations

Description: This chapter introduces the essential components of computer programming and programming languages.

Detailed contents:

- Basic architecture of computers
- Computer programming
- Programming paradigms
- Types of programming languages

2. Design of programs

Description: This chapter focuses on the internal design of programs, paying special attention to the concept of algorithm.

Detailed contents:

- Computer algorithms
- Analysis of algorithms
- Data structures

3. Coding

Description: Acquiring knowledge on coding by using an imperative programming language.

Detailed contents:

- Program data
- Operators
- Advanced data structures
- Program statements
- Subprograms

4. Testing and debugging

Description: Learning principles and techniques about testing, debugging and deploying computer programs.

Detailed contents:

- Compilation-execution cycle
- Testing techniques
- Debugging techniques

LEARNING ACTIVITIES AND METHODOLOGY

- 1) Theoretical lectures: 1,5 ECTS. Página 1 de 2 Lectures oriented to present the theoretical concepts on programming.
- 2) Practical lectures: 1,5 ECTS. Classes in computer labs oriented to learn the use of an IDE and put into practice the syntax.
- 3) Programming exercises: 2,0 ECTS. Problem-based learning. Programming different pieces of code with the purpose of understanding those conceptual, technical, and methodological principles that underlie structured programming.
- 4) Individual study: 1,0 ECTS. Self-studying to prepare for partials and final exams

ASSESSMENT SYSTEM

- Midterm exam on programming foundations: 20%
- Midterm exam on tracing code: 20%
- Final exam: 60%

There is a minimum mark required on the final exam of 3.0 of 6

| | |
|---|----|
| % end-of-term-examination: | 60 |
| % of continuous assessment (assignments, laboratory, practicals...): | 40 |

BASIC BIBLIOGRAPHY

- Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers How to Think Like a Computer Scientist: Learning with Python 3, <https://media.readthedocs.org/pdf/howtothink/latest/howtothink.pdf> , 2018
- Ravi Sethi. Programming Languages. Concepts and Constructs., ADDISON-WESLEY..
- Stephenson, Ben. The Python Workbook, Springer, 2014

ADDITIONAL BIBLIOGRAPHY

- Allen B. Downey Think Python, O'Reilly Media, Inc., 2012
- Bill Lubanovic Introducing Python, O'Reilly Media, Inc., 2014
- George W. Gorsline. Computer Organization: Hardware Software., PRENTICE HALL INTERNATIONAL EDITIONS..
- González Duque, R. Python para todos, <http://mundogeek.net/tutorial-python/>.
- Guido van Rossum and the Python Development Team Python Tutorial Rel. 3.7.0. (tutorial oficial de Python), <https://docs.python.org/3/tutorial/> , 2017
- Stephen D.Burd. System Architecture. Hardware and Software in Business Information Systems., BOYD AND FRASER PUBLISHINGCOMPANY..