Programming

Academic Year: (2020 / 2021) **Review date: 10-07-2020**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: ALVAREZ RODRIGUEZ, JOSE MARIA

Type: Basic Core ECTS Credits : 6.0

Year : 1 Semester : 2

Branch of knowledge: Engineering and Architecture

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

-Experience in the use of computers will be valuable.

OBJECTIVES

Basic competences:

CB1. Students have demonstrated knowledge and understanding in a field of study that builds upon their general secondary education, and is typically at a level that, whilst supported by advanced textbooks, includes some aspects that will be informed by knowledge of the forefront of their field of study

CB2. Students can apply their knowledge and understanding in a manner that indicates a professional approach to their work or vocation, and have competences typically demonstrated through devising and sustaining arguments and solving problems within their field of study

CB3. Students have the ability to gather and interpret relevant data (usually within their field of study) to inform judgments that include reflection on relevant social, scientific or ethical issues

CB4. Students can communicate information, ideas, problems and solutions to both specialist and non-specialist audiences

CB5. Students have developed those learning skills that are necessary for them to continue to undertake further study with a high degree of autonomy

General competences:

CG2. Learn new methods and technologies from basic scientific and technical knowledge, and being able to adapt to new situations.

CG3. Solve problems with initiative, decision making, creativity, and communicate and transmit knowledge, skills and abilities, understanding the ethical,
social and professional responsibility of the engineering activity. Capacity for leadership, innovation and entrepreneurial spirit.

CG5. Use the theoretical and practical knowledge acquired in the definition, approach and resolution of problems in the framework of the exercise of their profession.

CE3. Use and program computers, operating systems, databases and software with application in engineering, and implement numerical algorithms in low and high level languages.

CT1. Work in multidisciplinary and international teams as well as organize and plan work making the right decisions based on available information, gathering and interpreting relevant data to make judgments and critical thinking within the area of study.

RA1. To have acquired sufficient knowledge and proved a sufficiently deep comprehension of the basic principles, both theoretical and practical, and methodology of the more important fields in science and technology as to be able to work successfully in them;

RA2. To be able, using arguments, strategies and procedures developed by themselves, to apply their

knowledge and abilities to the successful solution of complex technological problems that require creating and innovative thinking;

RA3. To be able to search for, collect and interpret relevant information and data to back up their conclusions including, whenever needed, the consideration of any social, scientific and ethical aspects relevant in their field of study;

RA6. To be aware of their own shortcomings and formative needs in their field of specialty, and to be able to plan and organize their own training with a high degree of independence.

According to "the common European framework for ICT Professionals in all industry sectors":

B.1. Application Development Level 3
B.2. Component Integration Level 2
B.3. Testing Level 2
B.5. Documentation Production Level 3
C.4. Problem Management Level 3

DESCRIPTION OF CONTENTS: PROGRAMME

1 Basic Concepts.
1.1 Structure of a computer system: hardware and software. Information encoding.
1.2 Bool algebra.
1.3 The notion of programming language. Programming paradigms: structured programming.
1.4 Basic definitions: algorithm, program, process, etc.
1.5 Compilation, debugging and execution processes.
1.6 Pseudocode.

2 Basic elements of programming.
2.1 Identifiers.
2.2 Variables and constants.
2.3 Simple datatypes.
2.4 Operators, expressions and statements.

3 Control flow.
3.1 Conditional statements.
3.2 Loop statements.
3.3 Other control flow statements.

4 Data structures and user-defined datatypes.
4.1 Definition and design principles.
4.2 Strings: concept, management and application.
4.3 Arrays: concept, management and application.
4.4 Pointers: concept, management and application.
4.5 Structs: concept, management and application.

5 Subprograms: procedures and functions
5.1 Definition and design principles.
5.2 Function signature, parameters and invocation.
5.3 Introduction to recursive functions.

6 Basic algorithms.
6.1 Searching and sorting.

7 Resource management.
7.1 Static vs dynamic memory
7.2 Memory basic operations: allocation and free.

8 Input/Output system.
8.1 File definition, use and types: text and binary.
8.2 File management: create, write, read and delete operations.

LEARNING ACTIVITIES AND METHODOLOGY

Theoretical Lectures: 1 ECTS

Practical Lectures: 1 ECTS
- Exercise resolution

- Best coding practices
- General tutoring

Team Work: 3 ECTS
- Project design and development
- Application of best coding practices

Individual Work: 1 ECTS
- Contribution to team project
- Study and preparation of exams

ASSESSMENT SYSTEM

CONTINUOUS EVALUATION (40%)
- Mid-term exam: 20%
- Questionnaires of continuos evaluation: 10%
- Mid-term delivery of the final project: 10%

FINAL EVALUATION (60%)
- Final project: 30%
- Final exam: 30%

A minimum grade of 5.0 both in the final project and in the final examination is required to pass the course.

% end-of-term-examination:	60
% of continuous assessment (assigments, laboratory, practicals…):	40

BASIC BIBLIOGRAPHY

- Allen B. Downey Think Python: How to Think Like a Computer Scientist, 2nd edition, O'Reilly, 2015
- Andrew Koenig C Traps and Pitfalls, Addison-Wesley Professional, 1989
- Anthony Scopatz, Kathryn D. Huff Effective Computation in Physics: Field Guide to Research with Python, O'Reilly, 2015
- Brian W. Kernighan / Dennis Ritchie The C Programming Language, Pearson, 2015
- David M. Beazley Python Cookbook: Recipes for Mastering Python 3, O'Reilly, 2011
- Jose María Alvarez Rodríguez Hands on Programming with Python: Theory and Practice, Amazon KDP Publishing, 2020
- K. N. King C Programming: A Modern Approach, 2nd Edition, W. W. Norton & Company, 2008
- Luciano Ramalho Fluent Python, O'Reilly, 2015
- Mark Lutz Learning Python, 0'Reilly, 2013
- Paul Barry Head-First Python, 2nd edition, O'Reilly, 2016
- Robert C. Martin Clean Code: A Handbook of Agile Software Craftsmanship , Prentice Hall, 2008
- Samuel P. Harbison, Guy L. Steele Jr. C: A Reference Manual, 5th Edition, Pearson, 2002
- Zed A. Shaw Learn Python 3 the Hard Way, Addison-Wesley, 2016

ADDITIONAL BIBLIOGRAPHY

- Peter Prinz, Tony Crawford C in a Nutshell, O'Reilly Media, 2015
- Richard M. Reese Understanding and Using C Pointers, O'Reilly Media, 2013

BASIC ELECTRONIC RESOURCES

- Jose María Alvarez Rodríguez . Hands on Programming with Python: https://chemaar.github.io/python-programming-course/