Programming

Academic Year: (2023 / 2024) **Review date: 13-06-2023**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: FRAGA VAZQUEZ, ANABEL

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

OBJECTIVES

The goal of this course is to introduce basic programming techniques useful for telecommunication applications. The programming language is Java. However, the techniques taught in the course are generic, i.e. they apply to other programming languages as well. Upon successful completion of the course, the student will be able to:
- Understand the fundamentals of the imperative programming
- Understand the fundamentals of structured programming
- Understand and manage language data types
- Understand and manage the structures of flow control
- Know Java syntax
- Know program testing and debugging
- Analyze and understand Java programs
- Localize and fix syntax and functional errors
- Analyze and understand algorithms written in pseudocode or in flow diagrams
- Design and develop simple algorithms based on given prerequisites
- Develop simple Java programs from flow diagrams, pseudocode or functional descriptions
- Have knowledge of basic search and sorting algorithms
- Design, develop and execute program test plans.

DESCRIPTION OF CONTENTS: PROGRAMME

PART I: Programming fundamentals
- Essentials of computer architecture
- Programming languages
- Compiling and running code
- Elements of a program: data and algorithms
- Basic programming tools: algorithms, flow diagrams, and pseudo code.
- JAVA language.

PART II: Structured programming
- Primitive data types and operators
- Flow control
- Conditional sentences
- Loops

PART III: Introduction to program testing

PART IV: Advanced concepts
- The String class
- Arrays
- Input/output

PART IV: Modular programming
- Organizing code in methods
- Information Exchange between functions: passing parameters

LEARNING ACTIVITIES AND METHODOLOGY

The teaching methodology includes:
1. Lectures presenting theoretical knowledge. Basic textbooks for both theory and problems will also be recommended, which will allow the students to complete and deepen the subjects in which they are most interested.
2. Practice in computer labs, during which the students will develop and analyze programs using the theoretical concepts taught in lectures. The assignments are developed in groups, in order to promote teamwork.
3. Problem solving both on paper and with the computer, targeting self-evaluation.
4. Individual practical assignments in computer labs based on the assignments developed in groups.
5. Sharing the problems solutions and joined correction in order to develop the capacity of analyzing and communicating information relevant to problem-solving. Additionally, this activity will promote the change of critical opinions between the professor and the students and among students.
6. Use of new e-learning technologies with a known platform (edX) where students will have the opportunity to reinforce their learning based on: Cross-evaluation, self-learning, and video formats to allow the students to gain more knowledge at any time they want.

ASSESSMENT SYSTEM

The evaluation will be distributed throughout the term, and the final grade will consist of the following parts:
Continuous evaluation:
- Midterm lab exams (mandatory, individual): 40%. Two midterm programming exams on a computer.
- Programming project and its defense (mandatory, in a group of maximum 2): 20%. The students' solution to a programming project will be evaluated, focusing on the modular organization of the code through the use of methods.
- e-learning (individual): 10% [40-40-20]. Submission of programming exercises (40%), multiple choice exams (40%), analysis, and code understanding (20%).

End-of-term examination:
- Final exam (mandatory, individual): 30%. In the exam the knowledge acquired by the student will be evaluate: it is required to obtain at least 5/10 of the mark to fulfill the requirements of the continuous evaluation process. Likewise, students who have not followed the continuous assessment will be allowed to take a final exam with a value of 60% of the subject.
In the final exam, a minimum mark will be required (5 out of 10 points) to fulfill the continuous evaluation process's requirements (pass the subject). In the extraordinary evaluative process, the final mark will be the best one between the 100% of the final exam or the percentages applied in the continuous evaluation (according to the assessment legislation adopted by the University on May 31, 2011).
It will not be possible to present any kind of practical exercise in the extraordinary exam period.

% end-of-term-examination: 30
% of continuous assessment (assignments, laboratory, practicals¿): 70

% end-of-term-examination:	30
% of continuous assessment (assigments, laboratory, practicals…):	70

BASIC BIBLIOGRAPHY

- Alfonso Jiménez Marín, Franciso Manuel Pérez Montes Aprende a Programa con Java, Ediciones Paraninfo, 2016

- BRUCE ECKEL "Piensa en Java / Thinking in Java", Pearson Educación. Versión española: 2002 - 2ª Edición. Versión inglesa: 2007 - 4ª Edición.
- C. T. Wu Introduction to Object-Oriented Programming with Java 5th edition, McGraw-Hill, 2009.
- Herbert Schildt Java 9, ANAYA MULTIMEDIA, 2018
- RUSSEL WINDER & GRAHAM ROBERTS "Developing Java software", Wiley. 2006 - 3º Edición.

ADDITIONAL BIBLIOGRAPHY

- Donald E. Knuth The Art of Computer Programming, Addison-Wesley Educational Publishers Inc, 2011

BASIC ELECTRONIC RESOURCES

- Bert Bates; Kathy Sierra . Head First Java, 3rd Edition: https://bibliotecas.uc3m.es/permalink/f/1t7u60p/TN_cdi_safari_books_9781492091646