

Programming

Academic Year: (2020 / 2021)

Review date: 10-07-2020

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: BELLUCCI , ANDREA

Type: Basic Core ECTS Credits : 6.0

Year : 1 Semester : 1

Branch of knowledge: Engineering and Architecture

OBJECTIVES

The objective of this course is to introduce basic programming techniques useful for telecommunication applications. The programming language is Java. However, the taught techniques are generic, i.e. they apply to other programming languages as well.

The knowledge acquired by the students in this class contains:

- the fundamentals of the imperative programming
- the fundamentals of structured programming
- language data types
- Java syntax
- Program testing and debugging

At the end of the class the students will acquire both specific and generic skills.

The specific skills targeted by this class consist of:

- Analyze and understand Java programs
- Localize and fix syntax and functional errors (PO b)
- Analyze and understand algorithms written in pseudocode or in flow diagrams (PO a)
- Design and develop simple algorithms based on given prerequisites (PO a,c)
- Develop simple Java programs from flow diagrams, pseudocode or functional descriptions (PO a, c, k)
- Have knowledge of basic search and sorting algorithms (PO k)
- Design, develop and execute program test plans. (PO b,k)

The generic skills refer to:

- Capacity to solve complex problems by decomposition (PO e)
- Team work and work distribution for solving complex problems (PO d)
- Project development scheduling (PO k)
- Relevant information searching and selection (PO a)
- Ability to communicate information in speech, presentation and writing in the different activities (exercises, debates, practical assignments, etc.) (PO g)
- Students should have demonstrated knowledge and understanding in an area of study part of the basis of general secondary education, and is typically at a level that, whilst supported by advanced textbooks, includes some aspects that imply knowledge of the forefront of their field of study. (CB1)
- The students can apply their knowledge to their work or vocation in a professional manner and have competences typically demonstrated through devising and sustaining arguments and solving problems within their field of study . (CB2)
- Knowledge of raw materials and technologies, enabling him to learn new methods and technologies as well as to develop versatility to adapt to new situations . (CG3)
- Basic knowledge of the use and programming of computers , operating systems, databases and software with applications in engineering. (CG11)

The student is expected to:

- Have a critical attitude with respect to different solutions to a concrete problem
- Collaborate and cooperate with team members
- Employ the bibliography and other information sources for enlarging his knowledge

DESCRIPTION OF CONTENTS: PROGRAMME

This course aims to introduce students to foundational programming techniques. The programming language that will be used in the course is Java, although students will acquire broad skills that can be applied to other programming languages.

CONTENT:

This course aims to introduce students to foundational programming techniques. The programming

language that will be used in the course is Java, although students will acquire broad skills that can be applied to other programming languages.

1. Programming fundamentals: acquire the foundations of computer programming, focusing on the concept of algorithm
 - 1.1. Essentials of computer architecture
 - 1.2. Programming languages
 - 1.3. Algorithms and pseudo code
2. The Java programming language: elements and syntax of the language.
 - 2.1. Java characteristics and features
 - 2.2. The Java Virtual Machine (JVM)
 - 2.3. Primitive data types and operators
 - 2.4. Flow control
 - 2.4.1. Conditional sentences
 - 2.4.2. Loops
 - 2.5. Java class library
 - 2.6. Monodimensional and multidimensional arrays
 - 2.7. Métodos
3. Programming environment. Analysis of tools and mechanisms that support the development of applications.
 - 3.1. The configuration of the environment
 - 3.2. Programming cycle
 - 3.3. Common programming errors

LEARNING ACTIVITIES AND METHODOLOGY

The teaching methodology includes:

1. Lectures presenting the theoretical knowledge. Basic text books for both theory and problems will be also recommended, which will allow the students to complete and deepen the subjects, in which they are most interested. (PO a,c,e)
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 the team work. (PO b,c,e,d)
3. Problem solving both in paper and with the computer, targeting the self-evaluation. (PO a,e)
4. Individual practical assignment in computer labs based on the assignments developed in groups. (PO e,g)
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. (PO d,g)
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. (PO a,b,c,e,d)

ASSESSMENT SYSTEM

The evaluation will be distributed throughout the term and the final grade will consist of the following parts:

- Programming final assignment (mandatory): 20%. (PO b,c,d,e,g,k)
- Individual quizzes and lab examinations: 20%. The student will have to solve with a computer and in a reasonable time a practical problem, with a complexity similar to the ones solved in the computer labs. (PO b,c,d,e,k)
- Final examination: 30%. (PO a,c,e,g)
- e-learning examination: 30% [40-40-20] (PO a,b,c,d,e)

In the final exam, a minimum mark will be required (5 out of 10 points) to fulfill the requirements of the continuous evaluation process. 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:	50
% of continuous assessment (assignments, laboratory, practicals...):	50

BASIC BIBLIOGRAPHY

- Donald E. Knuth The Art of Computer Programming, Addison Wesley, 2011
- Herbert Schildt Java: A Beginner's Guide (Sixth Edition), McGraw-Hill Education, 2014