

## Programming

Academic Year: ( 2019 / 2020 )

Review date: 09-05-2018

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

## REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

No requirements.

## OBJECTIVES

The objective of this class 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

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. (PO a)
- 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)

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.

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. Functions
3. Programming environment. Analysis of tools and mechanisms that support the development of applications.
  - 3.1. 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,d,e,k)
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,k)
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)

## ASSESSMENT SYSTEM

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

- Programming assignments: 16%. (PO b,c,d,e,g,k)

- Midterm examination: 8%. (PO a,c,e,g)
- Individual quizzes and computer examination: 24%. 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: 32%. (PO a,c,e,g)
- e-learning examination: 20% [40-40-20] (PO a,b,c,d,e)

In the final exam a minimum mark will be required (4 over 10 points) to fulfill the requirements of 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 with the assessment legislation adopted by the University on May 31, 2011).

It will not be possible to present any kind of practical exercises in June exams period.

- Final examination: 40%. It is required to pass 4/10 of the total score of this exam. (PO a,c,e,g)
- A final examination of 60% will be available for students who have not completed the continuous assessment.

Extraordinary call will follow the rules below:

- For students who followed the continuous assessment process, the extraordinary examination has the same weight as in the ordinary call. The final score is computed as the arithmetic sum.
- If the student did not follow the continuous assessment process is entitled to examination of the total score, i.e. 100%.
- Although the student had followed the process of continuous assessment, will right to choose option b), when it appears more favorable.

<b>% end-of-term-examination:</b>	40
<b>% of continuous assessment (assignments, laboratory, practicals...):</b>	60

#### BASIC BIBLIOGRAPHY

- C. T. Wu An Introduction to Object-Oriented Programming with Java 5th edition, McGraw-Hill, 2009.
- Weiss, Mark Allen Data structures and problem solving using Java , Pearson/Addison Wesley, 2010

#### ADDITIONAL BIBLIOGRAPHY

- R. Winder, G. Roberts Developing Java software, Wiley. 2006.
- Bruce Eckel Thinking in Java, Prentice Hall, 2000