COURSE DATA

Data Subject	
Code	34796
Name	Computer Programming
Cycle	Grade
ECTS Credits	6.0
Academic year	2018 - 2019

Study (s)

Degree	Center	Acad. year	Period
1402 - Degree in Telecommunications Electronic Engineering	School of Engineering	2	Second term

Subject-matter

Degree	Subject-matter	Character
1402 - Degree in Telecommunications Electronic Engineering	9 - Programming	Obligatory

Coordination

Name	Department
BENAVENT GARCIA, MARIA ROSER	240 - Computer Science

SUMMARY

Programming has 6 ECTS, is taught in the second semester of the second course in Telecommunication Electronics Engineering Degree. Its purpose is to give students an introduction to the Java programming language and provide a broad view of different APIs for the development of network and distributed applications. After completing the course students should be able to develop applications using networked and properly distributed object-oriented, parameterized types, class hierarchies, concurrency and synchronization of concurrent tasks.

The general objectives are listed below:

- Develop applications using object-oriented concepts.
- Declaring and using appropriate class hierarchies, abstract classes, interfaces and parameterized types.
- Develop applications that use concurrency and shared resources to synchronize tasks using semaphores or monitors.
- Create input and output streams given some specifications.

- Develop networked applications using different protocols.
- Use integrated development environments for development, debugging and execution applications.
- Use appropriate tools to compile and run applications.

PREVIOUS KNOWLEDGE

Relationship to other subjects of the same degree

There are no specified enrollment restrictions with other subjects of the curriculum.

Other requirements

To a proper understanding of the subject matter it is recommended to have the background given in an introductory course in programming, as the provided in the subject Informatica.

OUTCOMES

1402 - Degree in Telecommunications Electronic Engineering

- G3 - Acquisition of the knowledge of the basic and technological subjects that allows students to learn new methods and theories and endows them with the versatility to adapt to new situations.
- G4 - Ability to solve problems with initiative, decision-making and creativity, and to communicate and transmit knowledge, abilities and skills, understanding the ethical and professional responsibility of the activity of a telecommunications technical engineer.
- R7 - Understand and use the basic principles of programming for telecommunication networks, systems and services.

LEARNING OUTCOMES

- Declare classes to create new types, creating objects and sending messages. Use the types defined by language (R7).
- Declare and use class hierarchies (R7).
- Declare and use parameterized types and use those provided by the language (R7).
- Declare and use threads for concurrent execution of code (R7).
- Explain and use semaphores and monitors to synchronize tasks when accessing shared resources (R7).
- List and use input/output streams. Use streams to add functionality to other low level streams (R3).
- Declare classes whose instances can be serialized (R7).
- Develop applications that use UDP: unicast and multicast (R7).
- Develop applications that use the TCP protocol using sockets (R7).
- Use integrated development environments for development, debugging and running applications (R2).

In addition, the student must develop the following skills:

- Find and interpret information in the Java API (G3).
- Discuss the advantages / disadvantages of different approaches to solve a given problem (G4).
- Develop, write and present academic texts. There will also be emphasized that any written text should have some quality standards based on cohesion, clarity, logic, … of the arguments and evidence contained therein (G4).

DESCRIPTION OF CONTENTS

1. Object Oriented Programming in Java

Review of concepts: classes, methods, objects, messages and encapsulation.
Reference vs. primitive types.
Inheritance, class hierarchies, abstract classes, interfaces, polymorphism
Parameterized types: declaration and use.
Exceptions: declaration and treatment.

2. Concurrent Programming

Concurrent tasks with threads
Problems accessing to shared resources: critical section
Synchronization mechanisms of concurrent tasks: semaphores and monitors

3. Input and Output

Input and output byte-oriented streams
Input and output character-oriented streams
Object serialization

4. Network and distributed programming

Applications based on UDP, TCP and HTTP

WORKLOAD

ACTIVITY	Hours	% To be attended
Theory classes	30,00	100
Laboratory practices	20,00	100
Classroom practices	10,00	100
Development of group work	10,00	0
Study and independent work	20,00	0
Readings supplementary material	10,00	0
Preparing lectures	10,00	0
Preparation of practical classes and problem	40,00	0
TOTAL	**150,00**	

TEACHING METHODOLOGY

The methodologies proposed for this subject are:

- Lectures: Development of the themes by providing a global and integrative vision, analyzing in greater detail the key aspects and more complex, encouraging at all times the participation of students.
- Practical classes in the classroom. They complement the expository lessons in order to apply the basic concepts and expand them with the knowledge and experience that they acquire during the execution of the proposed works. These classes solve problems and problems in the classroom. Tests will also be carried out by means of an app in mobile or computer type socrative on the knowledge acquired in the classroom by the students.
- Practical classes in the laboratory. Practices will be carried out individually or in small groups in which students will develop theoretical and practical contents through their application in realistic cases using the specific material and under the supervision of the teacher. In each practice, the students will perform an exercise prior to the practice that will be delivered before the practice. This exercise aims to analyze the work to be done in the laboratory, and relate it to the concepts seen in the theoretical classes. Attendance to internships is compulsory and is a non-recoverable activity.
- Autonomous student work. Realization of works, questions, problems outside the classroom, bibliographical searches, as well as the preparation of classes and exams (study). The realization of these activities will sometimes be individual, to empower the student's autonomy, and sometimes in small groups, to enhance the capacity for integration into working groups, as well as leadership and coordination capacity.
- E-learning platform. The e-learning platform (Aula Virtual) of the Universitat de València will be used as a communication support for students. Through it will have access to the didactic material used in class, as well as the problems and solved exercises.

EVALUATION

The evaluation of the course will consider the following dimensions:

- Continuous assessment based on participation and degree of involvement in the teaching-learning process, taking into account regular attendance at classroom activities (solved exercicis, Socrative tests), and individual workshops that will be proposed at the e-learning platform (N_Continua) (R2, R3).
- An intermediate individual continuous evaluation exam: midterm exam. It will contain theoretical issues and problems. This test eliminates provided material for the final examination if the score is equal to or greater than five, and the midterm exam's score will average with the final exam's score (Nproves) (R7).
- Evaluation of activities in the laboratory. Occasionally you may make oral presentations (individually and / or group) to evaluate the writing and presentation skills (N_Practicas). To evaluate the grade of each session, previous work will score 20% and development practice 80% (R7).

Final Score = 10% N_Continua + 60% (N_Pruebas) + 30% N_Practicas

All the individual exams' score (midterm and the final one) will be equal or greater than 4 to calculate the final score.

- Alternative assessment to the continuous assessment. At this assessment, the Final exam will contain both individual theoretical and practical issues and problems (N_Examen).

Final Score = 70% (NExamen) + 30% NPracticas

The N_Pruebas score will be greater than or equal than 4.

In the second call, there will be a test for theory / problems /laboratory. The final score in the second call will be obtained by one of the two proposed options at the first call.

"In any case, the evaluation system will be governed by the one establishes in the Regulation of Evaluation and Qualification of the University of Valencia for Degrees and Masters (https://webges.uv.es/uvTaeWeb/MuestraInformacionEdictoPublicoFrontAction.do?accion=inicio&idEdictoSeleccionado=5639)".

REFERENCES

Basic

- Java. Cómo Programar. P. J. Deitel y H. M. Deitel. Pearson Educación, Séptima edición, 2008
- Java Network Programming and Distributed Computing. David Reilly, Michael Reilly. Addison-Wesley

Additional

- Core Java 2. Volumen I. Cay S. Horstmann ; Gary Cornell, Prentice Hall, séptima edición, 2005
- Core Java 2. Volumen II. Cay S. Horstmann ; Gary Cornell, Prentice Hall, séptima edición, 2006