

**COURSE DATA****Data Subject**

Code	34888
Name	Computer Programming
Cycle	Grade
ECTS Credits	6.0
Academic year	2019 - 2020

Study (s)

Degree	Center	Acad. year	Period
1403 - Degree in Telematics Engineering	School of Engineering	3	First term

Subject-matter

Degree	Subject-matter	Character
1403 - Degree in Telematics Engineering	12 - Programming	Obligatory

Coordination

Name	Department
GUTIERREZ AGUADO, JUAN	240 - Computer Science

SUMMARY

The course "**Programming**" is a subject of the third year of the Degree in Telematics Engineering, which covers part of the compulsory subject *Programming*.

In this course, the concepts and skills acquired in the second year course "Amplificación de Informática" are extended. The Java programming language is introduced (object orientation, inheritance, parametrized types and concurrency); Input/Output; network programming with different protocols (UDP, TCP and HTTP); and, distributed programming with RMI.

The student should acquire the ability to develop applications that use all these concepts and technologies to meet the requirements.



PREVIOUS KNOWLEDGE

Relationship to other subjects of the same degree

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

Other requirements

It is recommended to have studied all the previous subjects in the fields of computer science and Programming .

OUTCOMES

1403 - Degree in Telematics Engineering

- R1 - Ability for self-learning of new knowledge and techniques appropriate for the conception, development and exploitation of telecommunications systems and services.
- 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.
- R2 - Ability to use communication and computer applications (offimatics, databases, advanced calculation, project management, visualization, etc.) to support the development and exploitation of telecommunications and electronics networks, services and applications.
- R3 - Ability to use computer tools to find bibliographic resources and information related to telecommunications and electronics.
- R7 - Understand and use the basic principles of programming for telecommunication networks, systems and services.
- E6 - Ability to design networks and telematic services architectures.
- E7 - Ability to programme networked and distributed telematic services and applications.

LEARNING OUTCOMES

This course allows for the following learning outcomes or skills:

1. To program applications using appropriately the object oriented approach (G3, G4, R1).
2. To declare and use class hierarchies, abstract classes, interfaces and parametrized types (G3, G4, R1).
3. Develop applications that use concurrency and shared resources that synchronize the tasks (G3, G4, R1, E7).
4. Construct input/output streams to meet some specification. Use object serialization (G3, G4, R1).
5. Use integrated development editors to develop, debug and execute applications (G3, G4, R1).



6. Use the tools to compile and execute applications (G3, G4, R1).
7. Find and interpret the information in the Java API (G3, G4, R1).
8. Develop network applications using UDP (G3, G4, R1, R7, E6, E7)
9. Develop network applications using TCP (G3, G4, R1, R7, E6, E7)
10. Develop network applications using HTTP (G3, G4, R1, R7, E6, E7)
11. Develop distributed applications using the distributed object middleware RMI and to explain what is the function of each element (G3, G4, R1, R7, E6, E7).
12. Develop distributed applications using properly the studied elements: concurrency, input/output/serialization, protocols, etc. in new contexts (G3, G4, R1, R7, E6, E7).

DESCRIPTION OF CONTENTS

1. Object orientation in Java

Background: classes, methods, objects, messages and .
References and primitive types.
Inheritance, type hierarchy, abstract classes, interfaces and polymorphism
Parametrized types
Exceptions

2. Concurrent programming

Concurrent task: threads
Critical section in shared resources
Task synchronization with monitors.

3. Input/Output

Binary oriented input and output streams.
Character oriented input and output streams.
Object serialization

4. Network programming

Classes for the protocol UDP.
Classes for the protocol TCP.
Classes for the protocol HTTP.

**5. Distributed programming and middleware**

Middleware

Distributed programming with RMI

WORKLOAD

ACTIVITY	Hours	% To be attended
Theory classes	30,00	100
Laboratory practices	20,00	100
Classroom practices	10,00	100
Study and independent work	20,00	0
Readings supplementary material	10,00	0
Preparation of evaluation activities	18,00	0
Preparing lectures	10,00	0
Preparation of practical classes and problem	30,00	0
Resolution of online questionnaires	2,00	0
TOTAL	150,00	

TEACHING METHODOLOGY

The methodologies that will be used in this subject are:

- Lectures.
- Problem solving and discussion.
- Laboratory sessions to practice the concepts
- Autonomous study.

The University of Valencia e-learning platform (*Aula Virtual*) will be used to support the communication with students. Through this platform the students will have access to course materials used in class as well as the problems and exercises to solve

EVALUATION**FIRST CALL**

The note of the theory part will be obtained as:

$$N_T = 0.4 * N_P1 + 0.4 * N_P2 + 0.1 * N_C + 0.1 * N_O$$



where:

N_P1: is the mark of the first continuous evaluation test.

N_P2: is the note of the second continuous evaluation test.

N_C: is the evaluation note for selected bulletins.

N_O: is the mark of the evaluation online.

At the end of each laboratory session a questionnaire will be passed with questions related to what was developed in that session or can be evaluated through academic work. The note of the laboratory part will be obtained as:

$$N_P = N_C$$

where:

N_C: is the average grade of the questionnaires and the works.

FINAL NOTE:

If the note N_T and the note N_P are greater than 4 the average will be done:

$$N_F = 0.7 * N_T + 0.3 * N_P$$

Otherwise, the subject will be suspended on first call.

The note N_T is recoverable however, the note N_P is not recoverable per the second call.

N_P1, N_P2 and N_C and N_O evaluate the following competencies: G3, G4, R7, E6 and E7

N_C evaluates the following competencies: G3, G4, R1, R7, E6 and E7

SECOND CALL

On the date set by the ETSE for the second call, a review will be held containing theoretical aspects, questions and practical aspects.

If N_E is greater than 4 the weighted average will be performed:

$$N_F = 0.7 * N_E + 0.3 * N_P$$

where

N_E = Exam Note

N_P = Note of the laboratory session questionnaires

N_E evaluate the following competencies: G3, G4, R7, E6 and E7

In any case, the system of evaluation will be ruled by the established in the Regulation of Evaluation and



Qualification of the University of Valencia for Degrees and Masters. (
http://www.uv.es/graus/normatives/2017_108_Reglament_avaluacio_qualificacio.pdf).

REFERENCES

Basic

- Java, cómo programar. Deitel y Deitel. 9 ed. 2012. Pearson Educación
- Core Java, Volume I--Fundamentals, Cay S. Horstmann, Gary Cornell, 8 ed, 2008, Prentice Hall
- Core Java, Volume II--Advanced Features, Cay S. Horstmann, Gary Cornell, 8th ed, 2008, Prentice Hall
- Java Network Programming and Distributed Computing, David Reilly, Michael Reilly, Addison Wesley, 2002
- API de Java: <http://docs.oracle.com/javase/7/docs/api/>

Additional

- C1 Documentación: <http://docs.oracle.com/javase/8/docs/>

ADDENDUM COVID-19

This addendum will only be activated if the health situation requires so and with the prior agreement of the Governing Council

English version is not available