



POLITÉCNICA

INTERNATIONAL
CAMPUS OF
EXCELLENCE

COORDINATION PROCESS OF
LEARNING ACTIVITIES
PR/CL/001



E.T.S. de Ingenieros
Informaticos

ANX-PR/CL/001-01

LEARNING GUIDE

SUBJECT

105000394 - Programming project

DEGREE PROGRAMME

10II - Grado En Ingenieria Informatica

ACADEMIC YEAR & SEMESTER

2018/19 - Semester 1

Index

Learning guide

1. Description.....	1
2. Faculty.....	1
3. Prior knowledge recommended to take the subject.....	2
4. Skills and learning outcomes	3
5. Brief description of the subject and syllabus.....	4
6. Schedule.....	6
7. Activities and assessment criteria.....	8
8. Teaching resources.....	11

1. Description

1.1. Subject details

Name of the subject	105000394 - Programming project
No of credits	3 ECTS
Type	Optional
Academic year of the programme	Third year
Semester of tuition	Semester 5
Tuition period	September-January
Tuition languages	English
Degree programme	10II - Grado en ingenieria informatica
Centre	10 - Escuela Tecnica Superior de Ingenieros Informaticos
Academic year	2018-19

2. Faculty

2.1. Faculty members with subject teaching role

Name and surname	Office/Room	Email	Tutoring hours *
Guillermo Roman Diez (Subject coordinator)	2304	guillermo.roman@upm.es	M - 10:00 - 14:00 Tu - 10:00 - 12:00 Please send an e-mail to set up an appointment before going to the instructor's office

Raul Alonso Calvo	2315	raul.alonso@upm.es	M - 10:00 - 13:00 W - 10:00 - 13:00 Please send an e-mail to set up an appointment before going to the instructor's office
-------------------	------	--------------------	--

* The tutoring schedule is indicative and subject to possible changes. Please check tutoring times with the faculty member in charge.

3. Prior knowledge recommended to take the subject

3.1. Recommended (passed) subjects

- Algoritmos y estructura de datos
- Programacion I
- Programacion II
- Programacion para sistemas
- Concurrencia

3.2. Other recommended learning outcomes

El plan de estudios Grado en Ingenieria Informatica no tiene definidos otros conocimientos previos para esta asignatura.

4. Skills and learning outcomes *

4.1. Skills to be learned

CG-1/21 - Capacidad de resolución de problemas aplicando conocimientos de matemáticas, ciencias e ingeniería.

CG-19 - Capacidad de usar las tecnologías de la información y la comunicación.

CG-2/CE45 - Capacidad para el aprendizaje autónomo y la actualización de conocimientos, y reconocimiento de su necesidad en el área de la informática.

CG-24/25/26/27 - Capacidad para trabajar en el contexto internacional, comunicándose en lengua inglesa y adaptándose a un nuevo entorno.

CG-3/4 - Saber trabajar en situaciones carentes de información y bajo presión, teniendo nuevas ideas, siendo creativo.

CG-5 - Capacidad de gestión de la información.

CG-6 - Capacidad de abstracción, análisis y síntesis

CG-7:10/16/17 - Capacidad para trabajar dentro de un equipo, organizando, planificando, tomando decisiones, negociando y resolviendo conflictos, relacionándose, y criticando y haciendo autocrítica

Ce 14/15 - Conocer el software, el hardware y las aplicaciones existentes en el mercado, así como el uso de sus elementos, y capacidad para familiarizarse con nuevas aplicaciones informáticas.

4.2. Learning outcomes

RA283 - Experiencia del desempeño profesional del ingeniero y de sus funciones más habituales en un entorno real de empresa.

RA284 - Capacitación para diseñar las líneas maestras de un proyecto.

RA521 - Resolver problemas algorítmicos no triviales Documentar clases y bibliotecas, tanto de manera pública

RA522 - Usar y definir estructuras de datos eficientes y adecuadas a cada problema

RA278 - Desarrollar la solución matemática y algorítmica mas apropiada a un problema informático que requiera un tratamiento especialmente complejo, analizando y exponiendo su viabilidad.

* The Learning Guides should reflect the Skills and Learning Outcomes in the same way as indicated in the Degree Verification Memory. For this reason, they have not been translated into English and appear in Spanish.

5. Brief description of the subject and syllabus

5.1. Brief description of the subject

This course aims at putting in practice the knowledge acquired by the students during the programming courses by means of the development of a programming project. The main goal of this course is to familiarize the students with the professional software development: the student will take design and programming decisions, and will work with techniques and tools used in software development companies.

The course will be based on the development of a software project in Java in groups of 3-4 students. To do so, multiple projects will be offered and the groups will be able to chose which project they are interested in developing. The students will put in practice the concepts explained in the theoretical sessions throughout the project development, such as how to face the design of the software, how to test their code, prepare the configuration scripts, document their code or manage the working in group issues.

5.2. Syllabus

1. Introducción to Software Development
2. Software Development Tools
 - 2.1. Version Control Systems
 - 2.2. Build Tools
3. Testing methodologies
4. Software design
 - 4.1. Abstraction-based software design
 - 4.2. Design patterns
5. Software Development
 - 5.1. Continuous Integration
 - 5.2. Test-Driven Development
 - 5.3. Iterative Development
6. Software Quality
 - 6.1. Technical Debt
 - 6.2. Documentation and code style
 - 6.3. Recording Activity: Logs

6. Schedule

6.1. Subject schedule*

Week	Face-to-face classroom activities	Face-to-face laboratory activities	Other face-to-face activities	Assessment activities
1	Introduction to Software development Duration: 02:00 Lecture			
2	Version Control Systems Duration: 02:00 Lecture			
3	Build tools Duration: 02:00 Lecture			
4	Testing methodologies Duration: 02:00 Lecture			
5	Testing methodologies Duration: 02:00 Lecture			
6		Laboratory Practice Duration: 02:00 Laboratory assignments		Laboratory practice Group work Continuous assessment Duration: 02:00
7	Abstraction-based software design Duration: 02:00 Lecture			
8	Design Patterns Duration: 02:00 Lecture			
9	Continuous Integration Duration: 02:00 Lecture			
10	Test-driven development Duration: 02:00 Lecture			
11	Test-driven development Duration: 02:00 Lecture			
12	Iterative Development Duration: 02:00 Lecture			
13	Software quality: Technical debt Duration: 02:00 Lecture			

14	Software quality: Documentation and code style Duration: 00:00 Lecture			
15	Software quality: Recording Activity, Logs Duration: 02:00 Lecture			
16				Written Exam Written test Continuous assessment Duration: 02:00 Programming Project Group work Continuous assessment and final examination Duration: 20:00 Written Exam Written test Final examination Duration: 02:00

The independent study hours are training activities during which students should spend time on individual study or individual assignments.

Depending on the programme study plan, total values will be calculated according to the ECTS credit unit as 26/27 hours of student face-to-face contact and independent study time.

* The subject schedule is based on a previous theoretical planning of the subject plan and might go through experience some unexpected changes along throughout the academic year.

7. Activities and assessment criteria

7.1. Assessment activities

7.1.1. Continuous assessment

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
6	Laboratory practice	Group work	No Presential	02:00	10%	0 / 10	CG-6 CG-5 CG-19 CG-7:10/16/17 CG-2/CE45 CG-1/21 CG-3/4
16	Written Exam	Written test	Face-to-face	02:00	20%	0 / 10	CG-6 CG-5 CG-19 CG-1/21 CG-3/4
16	Programming Project	Group work	No Presential	20:00	70%	0 / 10	CG-6 CG-5 Ce 14/15 CG-19 CG-24/25/26/27 CG-7:10/16/17 CG-2/CE45 CG-1/21 CG-3/4

7.1.2. Final examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
16	Programming Project	Group work	No Presential	20:00	70%	0 / 10	CG-6 CG-5 Ce 14/15 CG-19 CG-24/25/26/27 CG-7:10/16/17 CG-2/CE45 CG-1/21 CG-3/4

16	Written Exam	Written test	Face-to-face	02:00	30%	0 / 10	Ce 14/15 CG-6 CG-5 CG-19 CG-24/25/26/27 CG-7:10/16/17 CG-2/CE45 CG-1/21 CG-3/4
----	--------------	--------------	--------------	-------	-----	--------	--

7.1.3. Referred (re-sit) examination

Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
Written exam	Written test	Face-to-face	02:00	30%	0 / 10	CG-6 CG-5 Ce 14/15 CG-19 CG-24/25/26/27 CG-7:10/16/17 CG-2/CE45 CG-1/21 CG-3/4
Programming Project	Group work	Face-to-face	20:00	70%	0 / 10	CG-6 CG-5 Ce 14/15 CG-19 CG-24/25/26/27 CG-7:10/16/17 CG-2/CE45 CG-1/21 CG-3/4

7.2. Assessment criteria

Continuous evaluation:

The course have two assignments that must be

- A laboratory practice, which will be done in pairs of two students around weeks 5-6.
- A programming project, which will be done in groups of 3-4 students and must be delivered before the date of the final exam.

The final grade is divided in three parts:

- Laboratory practice (10%)
- Programming Project (70%)
- Written exam (20%)

Final exam:

Those students who are interested in "final exam evaluation", according to the rules described in <https://www.fi.upm.es/?pagina=1147&idioma=english>, must notify to the coordinator no later than 15 calendar days after the beginning of the subject lectures.

The final grade for these students will have two parts:

- Programming Project (70%)
- Written exam (30%)

Referred (re-sit) examination

The final grade for these students that did not pass the course will have two parts:

- Programming Project (70%)
- Written exam (30%)

The grade obtained on January in the programming project and in the written exam will be saved for the extraordinary examination done in July and the students can decide which parts of the course they want to repeat on July.

8. Teaching resources

8.1. Teaching resources for the subject

Name	Type	Notes
Moodle	Web resource	All material will be available in online