



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

2024/25 - Semester 2

Index

Learning guide

1. Description.....	1
2. Faculty.....	1
3. Prior knowledge recommended to take the subject.....	2
4. Skills and learning outcomes	2
5. Brief description of the subject and syllabus.....	3
6. Schedule.....	5
7. Activities and assessment criteria.....	7
8. Teaching resources.....	10
9. Other information.....	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 6
Tuition period	February-June
Tuition languages	English
Degree programme	10II - Grado en Ingenieria Informatica
Centre	10 - Escuela Tecnica Superior De Ingenieros Informaticos
Academic year	2024-25

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 - 12:00 - 15:00 W - 12:00 - 15:00 Please send an e-mail to set up an appointment
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

* 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

- El Arte De Programar
- ProgramaciÓn li
- Algoritmos Y Estructura De Datos
- ProgramaciÓn I
- Programacion Para Sistemas
- Concurrencia

3.2. Other recommended learning outcomes

The subject - other recommended learning outcomes, are not defined.

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

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

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

* 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. Software Quality
 - 3.1. Static Analysis Tools
4. Software Testing
 - 4.1. Test automation frameworks
5. Software Development
 - 5.1. Continuous Integration
 - 5.2. Agile methodologies
6. Software design

6. Schedule

6.1. Subject schedule*

Week	Type 1 activities	Type 2 activities	Distant / On-line	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	Software testing Duration: 02:00 Lecture			
5	Test automation frameworks Duration: 02:00 Lecture			
6	Test driven development Duration: 02:00 Lecture			
7	Laboratory Practice Duration: 02:00 Laboratory assignments			
8	Laboratory Practice Duration: 02:00 Laboratory assignments			Laboratory practice (NON RECOVERABLE) Group work Progressive assessment Not Presential Duration: 02:00
9	Software Development: Continuous Integration Duration: 02:00 Lecture			
10	Software Development: Agile Methodologies Duration: 02:00 Lecture			
11	Software Design Duration: 02:00 Lecture			
12	Software Design Duration: 02:00 Lecture			

13	Project tutorial Duration: 02:00 Laboratory assignments			
14	Project tutorial Duration: 02:00 Laboratory assignments			
15				
16	Written exam Duration: 01:00 Additional activities			Written Exam Written test Progressive assessment and Global Examination Presential Duration: 01:00 Programming Project Group work Progressive assessment and Global Examination Not Presential Duration: 20:00
17				

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 schedule is based on an a priori planning of the subject; it might be modified during the academic year, especially considering the COVID19 evolution.

7. Activities and assessment criteria

7.1. Assessment activities

7.1.1. Assessment

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
8	Laboratory practice (NON RECOVERABLE)	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	01:00	20%	4 / 10	CG-1/21 CG-3/4 CG-5 CG-6 CG-19
16	Programming Project	Group work	No Presential	20:00	70%	5 / 10	CG-1/21 CG-3/4 CG-7:10/16/17 CG-2/CE45 CG-6 CG-5 Ce 14/15 CG-19 CG-24/25/26/27

7.1.2. Global examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
16	Written Exam	Written test	Face-to-face	01:00	20%	4 / 10	CG-1/21 CG-3/4 CG-5 CG-6 CG-19
16	Programming Project	Group work	No Presential	20:00	70%	5 / 10	CG-1/21 CG-3/4 CG-7:10/16/17 CG-2/CE45 CG-6 CG-5 Ce 14/15 CG-19 CG-24/25/26/27

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	20%	4 / 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%	5 / 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

Progressive evaluation:

The course have two practical assignments that must be done along the course:

- A **laboratory practice**, which will be done in pairs of two students around weeks 7-8.
- A **programming project**, which will be done in groups of 3-4 students and must be submitted before the date of the final exam. The project will have two submission dates, one intermediate deadline (4 weeks before the final deadline) and the final deadline when the project must be completely finished.

The final grade of the course is divided in three parts:

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

Laboratory practice is non recoverable as its main goal the environment preparation and to be familiar with the tools that will be used in the project. For these reasons, it does not make sense to do the laboratory practice in the referred evaluation.

As the written exam is a multiple choice exam, for academic reasons, the solution of the exam will not be publish.

Global evaluation:

The global evaluation includes the same evaluation parts that the progressive evaluation. As the written exam and the project are done in the official evaluation period, there is no option to recover them in the global evaluation. Thus, the final grade for these students will have the same three parts than before:

- Laboratory practice (10%) - cannot be submitted again as it is non recoverable
- Programming Project (70%)
- Written exam (20%)

Referred (re-sit) examination

In the referred evaluation, students can submit their projects and have another chance to pass the written exam. The grades obtained in the laboratory practice, the programming project and in the written exam will be saved for the referred examination. Students can decide which parts of the course they would like to repeat on the referred evaluation. The final grade for the referred evaluation will have three parts:

- Laboratory practice (10%) - cannot be submitted again as it is non recoverable
- Programming Project (70%)
- Written exam (20%)

IMPORTANT: A system for detecting plagiarisms will be used and the groups involved in plagiarism will be penalized with a failing grade in the corresponding examination period according to the UPM norms.

8. Teaching resources

8.1. Teaching resources for the subject

Name	Type	Notes
Moodle	Web resource	All material will be available in online
Gitlab server	Equipment	A Gitlab server for working on the practical part of the course
SonarQube server	Equipment	A SonarQube server for the practical parts of the course
Practical TDD and Acceptance TDD for Java Developers	Bibliography	L. Koskela, Test Driven: Practical TDD and Acceptance TDD for Java Developers, Manning Publications Co., Greenwich, CT, USA, 2007.
Clean code: A handbook of agile software craftsmanship	Bibliography	Robert C. Martin, Clean code: A handbook of agile software craftsmanship, 1 ed., Prentice Hall PTR, Upper Saddle River, NJ, USA, 2008.
The art of agile development	Bibliography	James Shore and Shane Warden, The art of agile development, first ed., O'Reilly, 2007

9. Other information

9.1. Other information about the subject

There is no extra information.