1. BASIC INFORMATION

Course	Object-Oriented Programming
Degree program	Bachelor's Degree in Computer Engineering
School	School of Architecture, Engineering and Design
Year	First
ECTS	6
Credit type	Basic
Language(s)	Spanish / English
Delivery mode	Campus-based / Virtual
Semester	S1
Academic year	2025-2026
Coordinating professor	Pedro J. Lara Bercial

2. PRESENTATION

The aim of this course is to introduce the fundamental concepts of object orientation and their implementation using a programming language (Java in this case).

This course is essentially a continuation of the Fundamentals of Programming course, although to a certain extent it also builds on some of the knowledge acquired in the Fundamentals of Computer Science course.

3. COMPETENCIES AND LEARNING OUTCOMES

Knowledges

- **CON06** Basic knowledge of computer use and programming, operating systems, databases, and software programs applicable in engineering.
- **CON21** Knowledge of the theoretical foundations of programming languages and associated lexical, syntactic, and semantic processing techniques, and the ability to apply them to the creation, design, and processing of languages.

Skills

- **HAB03** Ability to design, develop, select, and evaluate computer applications and systems, ensuring their reliability, security, and quality in accordance with ethical principles and current legislation and regulations.

Competences

- **CP05** Ability to conceive, develop, and maintain computer systems, services, and applications using software engineering methods as a means to ensure their quality, according to the knowledge acquired as specified in the specific competences of the degree.

4. CONTENTS AND LEARNING OUTCOMES

Specific learning outcomes for the subject

Subject-Specific Knowledge

- Describe the basic concepts inherent in the object-oriented paradigm
- Describe what a software design pattern is, applying some as examples.

Subject-Specific Skills

- Manage program control through classes.
- Apply basic software engineering concepts to practically solve medium-scale projects.
- Utilize the fundamental concepts of object-oriented programming paradigm (class design, inheritance, dependencies, associations, multiplicity, access modifiers, method overloading, abstract classes, polymorphism, and interfaces) to implement programs in an object-oriented programming language.
- Properly utilize modeling tools and languages.
- Demonstrate proficiency in programming environments, independently identify and rectify errors at each stage, and use debugging techniques when necessary.

Contents

- Unit 1:
  - Introduction to Classes and Objects. Fundamental concepts of the object-oriented programming paradigm.
  - Attributes, Constructors, and Methods. Design and implementation of classes, including declaration and use of attributes and methods. Constructors, object creation, and the principle of encapsulation with access modifiers.

- Unit 2:
  - Inheritance and Collections. Introduction to inheritance as a mechanism for reuse and specialization to create class hierarchies. Definition and use of object collections and implementation of relationships between classes.

- Unit 3:
  - Advanced Class Design.Method overloading and overriding.
  - Abstract Classes, Polymorphism, and Interfaces.

- Unit 4:
  - Quality Assurance and Design Patterns.
  - Basic Graphical User Interfaces. Introduction to the development of Graphical User Interfaces (GUIs) in an object-oriented environment.

5. TEACHING-LEARNING METHODOLOGIES

The following are the types of teaching-learning methodologies that will be applied:

- Master class
- Problem-based learning
- Project-based learning
- Workshop-based learning

6. LEARNING ACTIVITIES

The following identifies the types of training activities that will be carried out and the number of hours each student will devote to each of them:

Campus-based mode:

Learning activity	Total Time	Class Time	Use of AI
Masterclass	12	12	Not allowed
Practical application master classes	18	18	Not allowed
Problem solving	30	9	Suggested
Research and projects	25	0	Allowed
Activities in workshops and/or laboratories	15	15	Allowed
Self-employment	44	0	Suggested
Debates and colloquiums	4	4	Not allowed
In-person assessment tests	2	2	Not allowed
TOTAL	**150 hours**	**60 hours (40%)**	

Virtual mode:

Learning activity	Total Time	Synchronous Time	Use of AI
Masterclass	12	0	Not allowed
Practical application master classes	18	18	Not allowed
Problem solving	30	0	Suggested
Project development	25	0	Allowed
Synchronous activities in virtual workshops and/or laboratories	15	15	Allowed
Study of content and supplementary documentation (Independent Work)	44	0	Suggested
Virtual forum	4	0	Not allowed
Virtual assessment tests	2	2	Not allowed
TOTAL	**150 hours**	**60 hours (40%)**	

(*) The teacher will inform in advance in which particular activities AI tools can be used

7. ASSESSMENT

The following is a list of the assessment systems and their weighting in the overall grade for the course:

Campus-based mode:

Assessment system	Weight
In-person evaluation activities	50
Case/problem solving	10
Performance evaluation	10
Research and projects	20
Laboratory/workshop notebook	10

Virtual mode:

Assessment system	Weight
Virtual evaluation activities	50
Case/problem solving	10
Performance evaluation	10
Projects	20
Virtual Laboratory/workshop notebook	10

On the Virtual Campus, when you access the course, you will be able to see the assessment activities you must complete in detail, as well as the submission dates and assessment procedures for each one.

7.1. Ordinary

To pass the course in the regular exam session, you must obtain a final grade (weighted average) of 5.0 or higher out of 10.0.

In any case, you must obtain a grade of 5.0 or higher on all assessment tests in order for it to be averaged with the rest of the activities. If you do not achieve 5.0 in any of them, the grade cannot exceed 4.0.

The professor reserves the right to request an additional test to any of the assessment tests if there are doubts about the student's authorship.

7.2. Extraordinary

To pass the course in the extraordinary exam session, you must obtain a final grade (weighted average) of 5.0 or higher out of 10.0.

In any case, you must obtain a grade greater than or equal to 5.0 in all assessment tests in order for it to be averaged with the rest of the activities. If you do not reach 5.0 in any of them, the grade cannot exceed 4.0.

The teacher reserves the right to request an additional test to any of the assessment tests if they have doubts about the student's authorship.

Activities that were not passed in the regular exam period must be resubmitted after receiving the corresponding corrections from the teacher, as well as those that were not submitted.

8. TIMELINE

This section indicates the timeline with delivery dates of assessment activities:

Assessment activities	Date
Participation Activities (individual and group)	Weeks 1-18
Knowledge Test (individual)	Week 10
Mini-project Submission (group) and Checkpoint	Weeks 17-18
Global Knowledge Test	Week 18

This schedule may be subject to change for logistical reasons related to the activities. Any changes will be communicated to students in a timely manner.

9. BIBLIOGRAPHY

The main recommended bibliography is indicated below:

- "UML y patrones : introducción al análisis y diseño orientado a objetos y al proceso unificado" / Craig Larman ; traducción, Begoña Moros Valle ; supervisión de la traducción y revisión técnica, Jesús García Molina. Disponible en Colección General (Campus Villaviciosa) (QA76.64.L37318 2008)

Additional recommended bibliography:

- Documentación oficial de Java: https://docs.oracle.com/javase/
- "Piensa en Java". Eckel, Bruce. Pearson Prentice Hall. Ed. 2008
- WikiBooks: Object Oriented Programming. https://en.wikibooks.org/wiki/Object_Oriented_Programming
- "El lenguaje unificado de modelado" / Grady Booch, James Rumbaugh, Ivar Jacobson; traducción José Sáez Martínez ; supervisión de la traducción y revisión técnica Jesús J. García Molina. Disponible en Colección General (Campus Villaviciosa) (QA76.64 .B66818 2004)

10. DIVERSITY AWARENESS UNIT

The Educational Guidance, Diversity, and Inclusion Unit (ODI) offers support to our students throughout their university life to help them achieve their academic goals. Other pillars of our work include the inclusion of students with specific educational support needs, universal accessibility on the university's various campuses, and equal opportunities.

This Unit offers students:

1. Support and monitoring through personalized counseling and plans for students who need to improve their academic performance.
2. In terms of diversity, minor adjustments are made to the curriculum, i.e., in terms of methodology and assessment, for students with specific educational support needs, thereby promoting equal opportunities for all students.
3. We offer students various extracurricular training resources to develop a range of skills that will enrich their personal and professional development.
4. Vocational guidance through the provision of tools and counseling to students who are unsure about their career path or who believe they have made the wrong choice of degree program.

Students who need educational support can write to us at:
orientacioneducativa@universidadeuropea.es

11. ONLINE SURVEYS

Your opinion matters!

The Universidad Europea encourages you to participate in several surveys which help identify the strengths an d areas we need to improve regarding professors, degree programs and the teaching-learning process.
The surveys will be made available in the “surveys” section in virtual campus or via e-mail.
Your assessment is necessary for us to improve.
Thank you very much for your participation.