Universidad de Alcalá

TEACHING GUIDE

Advanced Programming

Degree in
Telematics Engineering

Universidad de Alcalá

Academic Year 2025/2026

3rd Year - 1st Semester

Approved by the EPS School Board on June 16th, 2025

TEACHING GUIDE

Course Name:	**Advanced Programming**
Code:	**380001**
Degree in:	**Telematics Engineering**
Department and area:	**Automática** **Arquitectura y Tecnología de Computadores**
Type:	**Compulsory**
ECTS Credits:	**6.0**
Year and semester:	**3rd Year, 1st Semester**
Teachers:	María Dolores Rodríguez Moreno
Tutoring schedule:	Consultar al comienzo de la asignatura
Language:	Spanish/English Friendly

1. COURSE SUMMARY

Software development requires approaches that facilitate the creation of complex, reusable, and easily maintainable systems. To this end, it is essential to master programming models that allow for modular code organization and promote abstraction and encapsulation. Among these models, the object-oriented programming paradigm is widely used in both professional and academic settings.

This course focuses on the design and construction of applications using this paradigm, building on the knowledge acquired in previous courses such as Computer Systems and Programming. Through a theoretical and practical approach, students will develop the ability to analyze problems, model solutions using classes, and implement structured programs based on objects and their interactions.

The course is organized into four thematic blocks, covering both the theoretical foundations and the practical application of the concepts:

1. General concepts of object-oriented programming, presenting the differences between this paradigm and structured programming, as well as the fundamental elements of the object-oriented language.
2. Classes, methods, and inheritance, focusing on class design, use of methods and attributes, and the principles of encapsulation, inheritance, and polymorphism. Code organization through modules is also introduced.
3. Data structures and libraries are centered on the use of dynamic and flexible data structures, along with the integration of external and specialized libraries to extend the functionality of the developed software.
4. Exceptions and files, addressing mechanisms for error handling through exceptions, including the creation of custom exceptions, and exploring input/output operations with files and databases.

2. SKILLS

Basic, Generic and Cross Curricular Skills.

This course contributes to acquire the following basic, generic and cross curricular skills:

Specific Skills

This course contributes to acquire the following specific skills:

en_CT1 - Ability to learn independently new knowledge and techniques suitable for the conception, development or operation of telecommunication systems and services.

en_CT4 - Ability to analyze and specify the fundamental parameters of a communications system.

en_CTE7 - Programming capacity of services and telematic applications, in network and distributed.

Professional Competencies

This course provides the following professional competence(s) defined in section 5 of the Annex to Order CIN/352/2009:

CTE7 - Ability to program networked and distributed telematic services and applications.

Learning Outcomes

Upon successful completion of this course, students will be able to:

RA1. Describe the differences between the OOP paradigm and others.

RA2. Design, from the textual specification of a problem a set of related classes whose interaction provides a solution to the said problem.

RA3. Build OO programs using a specific programming language.

3. CONTENTS

Contents Blocks	Total number of hours
General concepts of OOP Differences between OOP and structured programming. Introduction to OOP elements. Language elements.	6 T + 6 P hours
Classes and inheritance Concept of classes, objects, methods, and attributes. Encapsulation, inheritance, and polymorphism. Special methods. Modules.	12 T + 12 P hours
Structures and Libraries Flexible data structures. External and specialized libraries.	6 T + 6 P hours
Exceptions and files Concepts of exceptions. Creation of custom exceptions. Working with files and DataBases.	4 T + 4 P hours

4. TEACHING - LEARNING METHODOLOGIES. FORMATIVE ACTIVITIES.

4.1. Credits Distribution

Number of on-site hours:	58 hours (56 hours on-site +2 exams hours)
Number of hours of student work:	92 hours (Includes study hours, preparation of activities, preparation of exams)
Total hours	150

4.2. Methodological strategies, teaching materials and resources

Face-to-face classes.	• Presentation and/or review of practical concepts. • Problem solving. • Laboratory sessions: aimed at consolidating the concepts explained in class through program implementation and the use of development tools. • Code execution and debugging. • Group practical activities.
Individual, group and online tutoring (forum, email, etc.)	• Review of practices and individualized progress monitoring. • Support for autonomous learning.
Independent work.	• Complementary readings and review of class materials. • Use of language models (LLMs) as a support tool to understand concepts, debug code, and generate object-oriented solutions.

5. ASSESSMENT: procedures, evaluation and grading criteria

Preferably, students will be offered a continuous assessment model that has characteristics of formative assessment in a way that serves as feedback in the teaching-learning process.

5.1. PROCEDURES

The evaluation must be inspired by the criteria of continuous evaluation (Learning Assesment Guidelines, LAG, art 3). However, in compliance with the regulations of the University of Alcalá, an alternative process of final evaluation is made available to the student in accordance with the Learning Assesment Guidelines as indicated in Article 10, students will have a period of fifteen days from the start of the course to request in writing to the Director of the Polytechnic School their intention to take the non-continuous evaluation model adducing the reasons that they deem convenient. The evaluation of the learning process of all students who do not apply for it or are denied it will be done, by default, according to the continuous assessment model. The student has two calls to pass the subject, one ordinary and one extraordinary.

5.2. EVALUATION

EVALUATION CRITERIA

The assessment criteria measure the level at which the competencies have been acquired by the student. For that purpose, the following are defined:

CE1. Identifies and understands the main concepts and ideas of object-oriented programming, distinguishing them from other programming paradigms.

CE2. Designs and implements classes, instances, and methods by applying encapsulation principles.

CE3. Employs inheritance and polymorphism mechanisms to structure class hierarchies and promote code reuse.

CE4. Identifies and applies key concepts related to data structures, adapting their use to the context of the program.

CE5. Organizes code in a modular way, using functions, modules, and libraries appropriately.

CE6. Applies special methods to extend the behavior of classes in a controlled and consistent manner.

CE7. Manages errors through exceptions, including the creation of custom exceptions, and works with input/output streams and databases.

CE8.Effectively integrates the concepts learned to develop functional, structured, and maintainable applications.

EVALUATION INSTRUMENTS

This section specifies the assessment tools used to evaluate each of the defined criteria.

- Laboratory Test (PRL): Exam conducted on a computer or on paper covering the content of the Laboratory Assignments. These represent 30% of the final grade.
- Midterm Exam (PEI): A midterm written test covering theoretical and practical content delivered in the first part of the course. It accounts for 30% of the final grade.
- Final Exam (PEF): A comprehensive written exam that evaluates the full scope of the course content. It carries a weight of 40% in the final grade.
- Laboratory Assignments (PL): Practical activities that students must complete and submit according to the schedule. These may include coding exercises, problem-solving tasks, use of libraries, or functional testing.

Ordinary call: continuous assessment/final exam

Continuous assessment will consist of two parts:

1. Practical exercises and/or intermediate exams: 60% of the final grade.

   The total number of activities proposed in this block will be communicated to the student during the introductory class. These activities include:

   * One Midterm Exam (PEI)

   * Laboratory Test (PRL)

2. Carrying out a theoretical-practical final exam (PEF) at the end of the teaching period: 40% of the final grade. The duration of this exam will be approximately 2 hours.

In addition, students must submit the Laboratory Assignments (PL) by the deadline set in a calendar provided at the beginning of the course. Submitting the PLs is a necessary condition to take the PRL.

Students who do not submit any PL or complete any test will be considered "not presented" (no-show) for the course.

Assessment through a final exam:

The final exam will consist of a single test on theoretical-practical contents that will constitute 100% of the course grade.
Students who have been granted an evaluation through the final exam will have the right to take this exam. The exam may be conducted orally and/or in writing.
The theoretical-practical contents include the practices from the subject's practice schedule. All practices submitted outside the established schedule will carry the penalty described in point 2 of the continuous evaluation.

Extraordinary call: final exam

Students who have not passed the subject in the ordinary call may opt for another extraordinary call consisting of a single test on theoretical-practical contents that will constitute 100% of the subject grade. The exam may be conducted orally and/or in writing.

The theoretical-practical contents include the practical assignments from the course.

GRADING CRITERIA

In the ordinary call-continuous assessment, the relationship between the competences, learning outcomes, criteria, and evaluation instruments is as follows.

Skills	Learning Outcomes	Evaluation criteria	Grading Tool	Contribution to the final mark
CTE1, CTE4, CTE7	RA1-RA3	CE1, CE2, CE3, CE4, CE8	PEI	30%
CTE1, CTE4, CTE7	RA1-RA3	CE2, CE3,CE4, CE5, CE6, CE7, CE8	PRL	30%
CTE1, CTE4, CTE7	RA1-RA3	CE1 - CE8	PEF	40%

In the ordinary call-final - final exam or the extraordinary call - the relationship between the competencies, learning outcomes, criteria, and evaluation instruments is as follows.

Skills	Learning Outcomes	Evaluation criteria	Grading Tool	Contribution to the final mark
CTE1, CTE4, CTE7	RA1-RA3	CE1-CE8	PEF	100%

The teaching-learning methodology and the assessment process will be adapted as needed, in accordance with the guidelines of the Diversity Support Unit, to implement curricular adaptations for students with specific needs.

6. BIBLIOGRAPHY

6.1. Basic Bibliography

- Object Oriented Programming with Python for Beginners: Mastering the Foundations of OOP. From Principles to Practice. 2 in 1 Guide. Autor: SAM CAMPBELL.
- Python Tutorial - Tapa blanda. GuidoVan Rossum (2012).

6.2. Additional Bibliography

Lenguajes de programación. Diseño e Implementación. Terence W.Pratt. Marvin V. Zelkowitz. Prentice Hall.

Disclosure Note

During the evaluation tests, the guidelines set out in the Regulations establishing the Rules of Coexistence of the University of Alcalá must be followed, as well as the possible implications of the irregularities committed during said tests, including the consequences for committing academic fraud according to the Regulation of Disciplinary Regime of the Students of the University of Alcalá.