Universidad de Alcalá

TEACHING GUIDE

Computer Science

Degree in
Industrial Technologies Engineering

Universidad de Alcalá

Academic Year 2025/2026

1st Year - 2nd Semester

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

TEACHING GUIDE

Course Name:	**Computer Science**
Code:	**610008**
Degree in:	**Industrial Technologies Engineering**
Department and area:	**Automática** **Computer Languages and Systems**
Type:	**Basic**
ECTS Credits:	**6.0**
Year and semester:	**1st Year, 2nd Semester**
Teachers:	Por definir
Tutoring schedule:	Consultar al comienzo de la asignatura
Language:	English

1. COURSE SUMMARY

Computer Science (Informática) is a basic training subject taught in the second semester, first year, of the Bachelor's Degree on Industrial Technology Engineering. It is the first subject related to computing in the degree.

In this course students will be introduced to the basic concepts related to the structure of current general purpose computers. Each element in this structure will be described from the point of view of the functionality it provides.

The subject will deal with the problems relating to the representation of information in digital systems, from the most elementary forms to some complex data structures, and also with representing actions and procedures in an algorithmic way, so they can be executed by a microprocessor.

The student will learn how to build programs using the basic tools provided by a general purpose programming language.

Requirements and Recommendations

No prerequisites are required.

2. SKILLS

Basic, Generic and Cross Curricular Skills.

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

en_CG2 - Ability to direct the activities object of the projects in the field of information technology in accordance with the knowledge acquired in accordance with the provisions of section 5, annex 2, of resolution BOE-A-2009-12977.

en_CG3 - Ability to design, develop, evaluate and ensure accessibility, ergonomics, usability and security of computer systems, services and applications, as well as the information they manage.

en_CG4 - Ability to define, evaluate and select hardware and software platforms for the development and execution of computer systems, services and applications, in accordance with the knowledge acquired as set out in section 5, annex 2, of resolution BOE-A-2009 -12977.

en_CG9 - Ability to solve problems with initiative, decision making, autonomy and creativity. Ability to know how to communicate and transmit the knowledge, skills and abilities of the profession of Computer Engineering Engineer.

en_CB1 - That students have demonstrated to possess and understand knowledge in an area of study that is based on general secondary education, and is usually found at a level that, although supported by advanced textbooks, also includes some aspects that involve knowledge from the forefront of their field of study.

en_CB2 - That the students know how to apply their knowledge to their work or vocation in a professional manner and possess the competencies that are usually demonstrated through the elaboration and defense of arguments and the resolution of problems within their area of study.

en_CB3 - That students have the ability to gather and interpret relevant data (usually within their area of study) to make judgments that include a reflection on relevant social, scientific or ethical issues.

en_CB4 - That students can transmit information, ideas, problems and solutions to both a

specialized and non-specialized public.

en_CB5 - That the students have developed those learning skills necessary to undertake further studies with a high degree of autonomy.

Specific Skills

This course contributes to acquire the following soecific skills:

en_CFB3 - Basic knowledge of the use and programming of computers, operating systems, databases and computer programs with application in engineering.

Learning outcomes:

- RAI1: To describe the structure of a current general-purpose computer and how its different elements interact with each other.
- RAI2. To express procedures for problems resolution in an algorithmic way.
- RAI3. To represent information to be stored and processed by a computer, from basic representations to complex data structures.
- RAI4. To handle different programming paradigms.
- RAI5. To use basic design techniques and software engineering.
- RAI6. To build programs using a programming language.
- RAI7. To justify the need for concurrent processes, the problems they cause and the available solutions.

3. CONTENTS

Unit 1. Introduction to computer science
Basic definitions. Functional structure. von Neumann machine. Machine and assembler languages. Compilers. Programs.

Unit 2. Information representation
Instructions, data, character codification, integers and floating point numbers. Rounding.

Unit 3. Introduction to operating systems
Processes. Memory, paging, virtual memory, file systems, Unix/Linux systems. Applications. Program execution in an OS.

Unit 4. Introduction to C programming language
Introduction to programming. Compilation and linking in C, variables, arithmetical expressions, control flow, Input/Output, constants, vectors and functions.

Unit 5. Types, operators and expressions
Variable names, data types, declarations, arithmetical operators, relational operators, increment and decrement operators, bitwise operators, assignment operators and expressions, conditional expressions, precedence and order of evaluation

Unit 6. Control flow
If-Else, Switch, For, While, Do-while, Break and Continue

Unit 7. Functions and program structure
Basics of functions, scope rules, function declaration and definition, variable types, argument passing by value and by reference.

Unit 8. Structures.
Basics of structures, structures and functions, arrays of structures, pointers to structures, unions.

Unit 9. Pointers and arrays.
Pointers and addresses, pointers and function arguments, pointers and arrays, pointer arrays, multi-dimensional arrays.

Unit 10. Dynamic memory
Dynamic memory allocation. Functions for dynamic memory allocation. Dynamically allocated arrays. Pointers to structures. Pointers as arguments in functions.

Contents Blocks	Total number of hours
Introduction to Computer Science, Information representation and Introduction to operating systems	6T + 6P
C language programming	24T + 20P

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
Total hours	150

4.2. Methodological strategies, teaching materials and resources

Theoretical classes (big groups): include oral presentations by the lecturer, problem solving, computer demonstrations.

Practical clases (lab work, small groups): Brief oral presentations by the instructor about practical issues in C programming, practical laboratory sessions: programming exercises in C, program creation, compilation, linking, debugging.

Tutorship (individual/groupal): doubt solving, autonomous learning support.

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.

EVALUATION INSTRUMENTS:

This section specifies which evaluation instrument will be applied to each evaluation criteria:

- The Partial Evaluation Tests (PEP) are tests about theoretical concepts and practical problems resolution.
- The Laboratory Tests (PL) consists on practical programming exercises with the computer.
- The Final Exam (PEF) consists on some theoretical questions about the whole subject, and practical programming exercises with the computer.

Evaluation by single exam and Extraordinary call

In these two cases, evaluation will consist of all the exercises of the Ordinary Continuous Evaluation Call together in one exam.

5.2. EVALUATION

EVALUATION CRITERIA:

Evaluation criteria must address the extent of skills acquisition by the student. For this purpose, the following ones are defined:

- CE1: The student has acquired technical knowledge about the different computer components, both physical and logical.
- CE2: The student shows ability and initiative in solving practical problems associated with programming.
- CE3: The student is able to use basic software tools to support code generation.
- CE4: The student demonstrates that he or she can analyze the execution of programs created by others.

GRADING CRITERIA:

The student must get at least 50% of the points to pass. As a general rule, if a student does not attend at least 60% of the total evaluation tests, they will be graded as “No Presentado"

Ordinary, Continuous Evaluation Call

Skill	Learning outcome	Evaluation criteria	Evaluation instrument	Weight
CFB3	RA1, RA3, RA7	CE1	PEI1	15%
	RA2, RA3, RA4, RA5, RA6	CE2, CE3	PL1	10%
	RA2, RA3, RA4, RA5, RA6	CE2, CE3	PL2	40%
	RA2, RA3, RA4, RA5, RA6	CE2, CE4	PEF	35%

Ordinary, Final Evaluation Call

Skil	Learning outcome	Evaluation criteria	Evaluation instrumen	Weight
CFB3	RA1–7	CE1–4	PEF	100%

Extraordinary Call

Same as Ordinary, Final Evaluation Call

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

- *El lenguaje de programación C*., B. Kernighan y D. M. Ritchie. Ed. Prentice Hall.
- *Curso de programación con C/C++*, F.J. Ceballos. Ed. RA-MA.
- *Introducción a la Informática*, Prieto, Lloris y Torres. Ed. McGraw-Hill

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.