Universidad de Alcalá

TEACHING GUIDE

Computer Systems

Degree in
Telecommunication Technologies Engineering (GITT)
Telecommunication Systems Engineering (GIST)
Telematics Engineering (GIT)
Electronic Communications Engineering (GIEC)

Universidad de Alcalá

Academic Year 2025/2026

1st Year - 1st Semester (GITT+GIST+GIT+GIEC)

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

TEACHING GUIDE

Course Name:	**Computer Systems**
Code:	**350003 (GITT+GIST+GIT+GIEC)**
Degree in:	**Telecommunication Technologies Engineering (GITT)** **Telecommunication Systems Engineering (GIST)** **Telematics Engineering (GIT)** **Electronic Communications Engineering (GIEC)**
Department and area:	**Automática** **Computer Architecture and Technology**
Type:	**Basic (GITT+GIST+GIT+GIEC)**
ECTS Credits:	**6.0**
Year and semester:	**1st Year - 1st Semester (GITT+GIST+GIT+GIEC)**
Teachers:	Mª del Mar Lendínez Chica
Tutoring schedule:	Check the e-learning publishing environment: https://uah.blackboard.com
Language:	English

1. COURSE SUMMARY

Computing Systems is a basic subject imparted in the first term of the first year in several Engineering degrees: Communication Electronics Engineering, Telecommunication Systems Engineering, Engineering on Telecommunication Technologies, and Telematics Engineering.

This subject aims to teach how to build computer programs using the basic tools provided by a general-purpose programming language. To achieve this goal, elementary concepts related to general-purpose computers must be addressed. Therefore, each constituent element will be described and justified from the perspective of its functionality. Likewise, the most relevant cases in information coding will be covered, starting with the simplest forms of data and progressing to medium-complexity data structures. Similarly, methods for coding actions and procedures in an algorithmic form will be presented.

The Programming subject, taught in the second term, is based on this course. Furthermore, Computing Systems provides essential knowledge for several subjects throughout the degree, including Algebra Lineal, Electrónica Digital, Diseño Electrónico, Sistemas Electrónicos Digitales, Sistemas Electrónicos Digitales Avanzados, Programación Avanzada, Programación Visual, and Sistemas Operativos. Some of these subjects are also offered in English.

In summary, this subject covers key concepts such as digital computer architecture, information coding, elements of the C language, standard input/output, functions, control statements, arrays, structures, arrays of structures, pointers, and dynamic memory allocation.

2. SKILLS

Basic, Generic and Cross Curricular Skills.

This course contributes to acquire the following generic skills, which are defined in the Section 3 of the Annex to the Orden CIN/352/2009:

en_TR2 - Knowledge of basic subjects and technologies that enables to learn new methods and technologies, as well as to provide versatility that allows adaptation to new situations.

en_TR3 - Aptitude to solve problems with initiative, decision making, creativity, and to communicate and to transmit knowledge, skills and workmanship, comprising the ethical and professional responsibility of the activity of the Technical Engineer of Telecommunication.

en_TR5 - Easy to handle specifications, regulations and mandatory standards.

en_TR8 - Capacity of working in a multidisciplinary and multilingual team and of communicating, both in spoken and written language, knowledge, procedures, results and ideas related to telecommunications and electronics.

en_TRU2 - Oral and written competencies.

en_TRU3 - Ability to manage information.

en_TRU4 - Autonomous learning skills.

en_TRU5 - Team work.

Professional Skills

This course contributes to acquire the following professional skills, which are defined in the Section 5 of the Annex to the Orden CIN/352/2009:

en_CB2 - Basic knowledge about the computers usage and programming, operating systems, databases and computer programs in engineering applications.

Learning Outcomes

After succeeding in this subject the students will be able to:

RA1. Interpret how simple programs work and binary operations.

RA2. Distinguish in a program the basic data types and control statements.

RA3. Identify the different elements use in the structure like functions, arrays and structures.

RA4. Program with pointers and control the application using dynamic memory

RA5. Code high difficulty programs in an autonomous way using the C language, applying directly to every work or scientific environment.

3. CONTENTS

Contents Blocks	Total number of hours
Fundamentals of computers	2 T + 2 P hours
Fundamentals of programming	14 T + 14 P hours
Structured data types	6 T + 6 P hours
Dynamic memory allocation	6 T + 6 P hours

Schedule

Weeks	Contents
01st	Concept of computers. Von Neumann digital computer architecture. Peripherals. Modern computers. Programs and programming languages. Information encoding. Operating systems. Exercises. Practice: The Integrated Development Environment (IDE).
02nd	Stages of program development. Introduction to the I/O concept, matrices, control statements, and functions. Primitive data types. Derived types. Type synonyms. Literals. Identifiers. Keywords. Comments. Exercises. Practice: Introduction to standard Input/Output, editing, compiling, executing, and debugging a simple program.
03rd	Variables. Symbolic constants. Numeric expressions. Operators. Precedence and order of evaluation. Type conversions. Exercises. Practice: Operators, binary operators, and basic control statements.
04th	Program structure. Inclusion and substitution directives. Declarations and definitions. Statements: simple and complex. Exercises. Practice: Operators, binary operators, and basic control statements.
05th	Functions: declaration, definition, and invocation. Passing arguments by value and by reference. Variable scope. Storage classes. Numeric data and character strings. Standard input and output streams. Formatted output. Formatted input. Character input and output. End-of-line and end-of-file characters. Input data validation. Exercises. Practice: Operators, binary operators, and basic control statements.
06th	Sequential and repetitive control statements. Exercises. Practice: Control statements.

07th	Arrays. One-dimensional numeric arrays. Associative arrays. Character strings. String manipulation functions. Array type and size. Multidimensional arrays. Exercises. Practice: Arrays and control statements.
08th	Arrays of character strings. Copying arrays. Handling blocks of bytes. Structures. Arrays of structures. Exercises. Practice: Arrays and control statements.
09th	Structures. Structures arrays. Exercises. Practice: Structured Datatypes.
10th	Creating pointers. Operators "address of" and "content of address". Operations with pointers. Pointers and arrays. Exercises. Practice: Structured datatypes manipulation.
11th	Pointers to arrays of characters. Arrays of pointers. Pointer to pointer. The array of pointers to arrays of characters. Sorting arrays of character strings. Exercises. Practice: Structured Datatypes. manipulation.
12th	Dynamic memory allocation. Functions for dynamic memory allocation. Exercises. Practice: Pointers and Dynamic Arrays.
13th	Dynamic arrays. Pointers to structures. Pointers as function parameters. Exercises. Practice: Pointers and Dynamic Arrays.
14th	Dynamic arrays of pointers. Pointers to pointers. Memory managing. Practice: Pointers and Memory managing.

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

The methodological strategy training is made up of the following types of activities:

Theoretical sessions

These sessions consist of lectures where the basic theoretical content is explained. Students must develop a programming mindset. Given the highly practical nature of the subject, theoretical sessions will include numerous practical examples.

All available classroom resources will be used, including the blackboard, overhead projector, and computer. These tools allow the lecturer to demonstrate the resolution and implementation of programs, enhancing the dynamics of the theoretical sessions.

Students will be encouraged to participate in class through dialogue, collective discussions, and exercises to promote collaborative learning. Additionally, they will be asked to actively engage by solving exercises on the board, allowing them to share and compare their solutions.

To monitor learning progress and maintain interest in programming, the lecturer may assign various activities for students to complete, either individually or in groups.

Lab sessions

The purpose of these sessions is to immerse students in the world of programming by guiding them through projects related to professional and employment contexts. For this purpose, the necessary computers and software for developing, compiling, executing, and debugging programs will be available.

Each student can develop the programs individually, but working in pairs is recommended.

At least PL will be assigned throughout the course. These activities will involve implementing programs of increasing difficulty, directly related to the content and examples covered in lectures.

The complexity and size of practical laboratory activities will gradually increase, allowing students to progressively develop relevant skills in the subject. Initially, practical exercises will be guided by pseudocode, helping students understand the correct steps for program execution. As they gain experience, the pseudocode guidance will be removed, giving them more freedom to write the code according to their own approach. The teacher will take on a mentoring role, guiding students throughout the process.

During the learning process, students may use both textbooks and online resources for research.

Tutoring

Tutoring schedules are established to support students who require assistance, either individually or in groups. They also allow the lecturer to monitor students' understanding of the concepts, enabling the adaptation of classes to enhance learning.

To complement these activities, students have access to a learning platform where they can find course-related information and resources, including:

- Documentation developed by the lecturers
- Practical exercises and materials
- Previous exam files
- Lecturer contact information and tutoring schedules
- Updated course information, including forums and announcements
- Exam schedules and grades

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.

Ordinary call

Several tests will be conducted throughout the term as part of the continuous assessment. Students who complete the final test will receive a grade, while those who do not take it will be marked as 'No submission'.

The single assessment model will consist of a single exam, which will take place on the same date as the final test in the continuous assessment. However, its format and content will differ. Both assessments may be conducted in written form or as a test on the Aula Virtual platform.

Extraordinary call

Students who do not pass the subject in the regular assessment will have another opportunity through the extra assessment. This model consists of a single exam covering questions related to all topics of the course.

5.2. EVALUATION

EVALUATION CRITERIA

The objective of this process is to assess the acquisition of subject competencies. To achieve this, the following criteria have been defined:

- **CE1.** The student has acquired the technical programming concepts explained during the course and applies them correctly.
- **CE2.** The student can solve problems accurately and conceptually, applying the theoretical topics covered in class. These problems may differ from those addressed in the practical sessions.
- **CE3.** The student can analyze and understand problems or technological needs in various fields, including business, industry, and science, providing solutions through efficient algorithms and programming.
- **CE4.** The student demonstrates proficiency in programming principles by correctly using data structures, control statements, pointers, and dynamic memory.
- **CE5.** The student shows the ability to understand, write, and modify complex programs. These programs consist of multiple functions and files.

GRADING TOOLS

This section presents the assessment tools applied according to the evaluation rules. All tests will be

conducted using the Moodle learning platform, hosted on the Web. These tests will consist of multiple-choice questions and short-answer exercises.

1. **Intermediate Assessment Test (PEI):** This test evaluates basic knowledge of the subject and includes questions covering half of the course topics. The PEI will take place in the laboratory and will consist of multiple-choice questions along with a programming task.
2. **Laboratory Test (PRL):** Up to four tests will be conducted, each based on activities performed during the practical lab sessions.
3. **Final Assessment (PEF):** This exam includes multiple-choice questions and short-answer exercises. The PEF will take place on the date set by the Escuela Politécnica Superior, either in the labs of the Automatic Department or in the designated classrooms.

GRADING CRITERIA

This section defines the assessment criteria required to pass the course.

Regular assessment. Continual evaluation.

In the regular assessment – continuous evaluation, the relationship between the rules, instruments, and grading criteria is summarized in the following table.:

Skill		Learning Outcomes	Evaluation criteria	Grading Tool	Contribution to the final mark
TRU4	CB2, TR2, TR3, TR8, TRU2	RA1, RA2, RA3	CE1-CE5	PEI	40%
	TR3, TR5, TR8, TRU2, TRU3, TRU5	RA1-RA5	CE1-CE5	PLn	20%
	CB2, TRU2, TRU3, TRU5, TR2, TR3, TR5, TR8	RA1-RA5	CE1-CE5	PEF	40% (100%)

To pass the course through continuous assessment, students must complete all tests proposed throughout the semester and submit all assignments by the specified deadline in Aula Virtual. If the PEF is not taken, the overall grade for the course will be marked as 'Not Presented' (NP).

Attendance to the laboratory:

Attendance to at least 80% of the laboratory sessions is mandatory. If a student has more than three unjustified absences, they will receive a grade of 'No Submission' in the regular assessment. During the practical sessions, students will develop the programs outlined in the documentation. They must identify and correct errors to achieve the expected results. Laboratory work is assessed through examinations, which are based on the content covered in the practical sessions. Passing only one of the three continuous assessment tests does not guarantee passing the course. The fact to pass just one of the three continuous tests does not mean passing the course.

Regular assessment, Final assessment

Skill	Learning Outcomes	Evaluation criteria	Grading Tool	Contribution to the final mark
CB2, TRU2, TRU3, TRU4, TRU5, TR2, TR3, TR5, TR8	RA1-RA5	CE1-CE5	PEF	100%

Extraordinary call

Skill	Learning Outcomes	Evaluation criteria	Grading Tool	Contribution to the final mark
CB2, TRU2, TRU3, TRU4, TRU5, TR2, TR3, TR5, TR8	RA1-RA5	CE1-CE5	PEF	100%

This exam will take place on the date scheduled by the Escuela Politécnica Superior for extraordinary sessions, either in the labs of the Automatic Department or in the designated classrooms. It will consist of a multiple-choice exam and programming questions requiring development.

The teaching-learning methodology and the evaluation process will be adjusted when necessary, following the guidelines of the Diversity Support Unit, to implement curricular adaptations for students with specific needs.

6. BIBLIOGRAPHY

6.1. Basic Bibliography

Basic bibliography in English:

- C How to Program, Deitel & Associates Inc. H.M.Deitel & P.J.Deitel

Basic bibliography in Spanish:

- Curso de programación con C/C++. Ed. RA-MA. Fco. Javier Ceballos.

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.