

Course guide

320002 - FI - Fundamentals of Informatics

Last modified: 15/09/2026

Unit in charge: Terrassa School of Industrial, Aerospace and Audiovisual Engineering
Teaching unit: 723 - CS - Department of Computer Science.

Degree: BACHELOR'S DEGREE IN CHEMICAL ENGINEERING (Syllabus 2009). (Compulsory subject).
BACHELOR'S DEGREE IN ELECTRICAL ENGINEERING (Syllabus 2009). (Compulsory subject).
BACHELOR'S DEGREE IN INDUSTRIAL ELECTRONICS AND AUTOMATIC CONTROL ENGINEERING (Syllabus 2009). (Compulsory subject).
BACHELOR'S DEGREE IN MECHANICAL ENGINEERING (Syllabus 2009). (Compulsory subject).
BACHELOR'S DEGREE IN TEXTILE TECHNOLOGY AND DESIGN ENGINEERING (Syllabus 2009). (Compulsory subject).
BACHELOR'S DEGREE IN INDUSTRIAL DESIGN AND PRODUCT DEVELOPMENT ENGINEERING (Syllabus 2010). (Compulsory subject).

Academic year: 2026 **ECTS Credits:** 6.0 **Languages:** Catalan, Spanish

LECTURER

Coordinating lecturer: Arratia Quesada, Argimiro Alejandro
López López, María José

Others: Marta Gatius Vila
Mugica Alvarez, Francisco José
Arratia Quesada, Argimiro Alejandro
López López, María José
Vellido Alcacena, Alfredo
Martín Prat, Ángela
Mylonakis Pascual, Nicolas Eduardo
López Herrera, Josefina
Fernández Durán, Pablo
Carlos Cano Domingo
Martha Ivon Cardenas Dominguez
Rene David Montero Nouel
Mateo Montón Velasco
Ivón Teresa Sánchez Cardenas

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:

CENG3-DIDP. Ability to program computers and operate systems, apply and use databases, and use software applications. (Basic training module).

CE03-INDUS. Basic knowledge of computer use and programming, operating systems, databases, and software programs applicable to engineering. (Basic training module)

Transversal:

CT05 N1. Effective use of information resources - Level 1. Identifying information needs. Using collections, premises and services that are available for designing and executing simple searches that are suited to the topic.

Basic:

CB1. That students have demonstrated possession and understanding of knowledge in a field of study that is based on general secondary education, and is typically found at a level that, while supported by advanced textbooks, also includes some aspects that involve knowledge from the forefront of their field of study.

CB2. That students can apply their knowledge to their work or vocation in a professional manner and possess the competencies typically demonstrated through the development and defense of arguments and problem-solving within their field of study.

TEACHING METHODOLOGY

- In-person sessions for content presentation.
- In-person practical work sessions.
- Independent study and completion of exercises.
- Preparation and completion of group-based assessed activities.

During content presentation sessions, the instructor will introduce the subject's theoretical foundations—concepts, methods, and results—illustrating them with suitable examples to facilitate understanding.

In-class practical work sessions will fall into three categories:

- a) Sessions where the instructor guides students through data analysis and problem-solving by applying techniques, concepts, and theoretical results.
- b) Sessions for students to present work completed in groups.
- c) Exam sessions.

Students are required to study independently to internalize concepts and solve assigned exercises, either manually or with the aid of a computer. Independent student work will be supported by online tools. Students will prepare group projects to be presented publicly during application sessions.

LEARNING OBJECTIVES OF THE SUBJECT

The main objective of the course is to teach students how to program in a high-level language. In order to pass the course, the student must be able to:

Understand the basic computing concepts related to hardware and software: computer architecture and operating systems.

Understand the fundamental concepts of computer programming.

Develop skills in the use of basic programming techniques and tools: algorithms and programs.

Design correct programs: well-structured, efficient, and readable.

Design data structures to represent the data of a given problem.

Carry out a medium-scale programming project at an industrial level.

Develop their capacity for abstraction through the use of programming patterns to solve real-world problems.

STUDY LOAD

Type	Hours	Percentage
Self study	90,0	60.00
Hours small group	60,0	40.00

Total learning time: 150 h

CONTENTS

TOPIC 1: INTRODUCTION TO COMPUTERS

Description:

- 1.1. Computer architecture.
- 1.2. Operating systems.
- 1.3. Computer programming.
- 1.4. Algorithms and programs.

Specific objectives:

For students to:

- Define the terms "hardware" and "software".
- Understand the general structure of a computer.
- Understand what an operating system is.
- Name and describe the various types of operating systems.
- Know what a program is.
- Name various programming languages.
- Name and describe the various programming paradigms.
- Understand the basic requirements of a program.
- Name and describe the phases of software development.
- Understand what an algorithm is.

Full-or-part-time: 11h

Laboratory classes: 2h

Self study : 9h

TOPIC 2: BASICS OF STRUCTURED PROGRAMMING

Description:

- 2.1. Structure of a program.
- 2.2. Objects.
- 2.3. Expressions and operators.
- 2.4. Elementary actions.
- 2.5. Writing instructions.

Specific objectives:

For students to:

- Properly use the objects in a program: constants and variables.
- Distinguish between correct and incorrect identifiers.
- Describe and use correctly the type of data available in the programming language as well as the defined operations.
- Describe the function of basic input and output actions and use them correctly.
- Understand assignment statements and use them correctly.
- Describe the structure of a program.
- Correctly develop a test suite.
- Determine whether the block structure of a program is correct.
- Correctly use alternative and iterative composition.
- Correctly build programs with objects, expressions, elementary actions and compositions.

Related activities:

- A type 1 activity, corresponding to an individual continuous assessment test during laboratory sessions.
- One or more type 2 activities, corresponding to an individual self-learning test.
- One or more type 3 activities, corresponding to an individual self-learning task.

Full-or-part-time: 28h

Laboratory classes: 10h

Self study : 18h

TOPIC 3: SUBROUTINES: ACTIONS AND FUNCTIONS

Description:

- 3.1. Actions.
- 3.2. Functions.
- 3.3. Parameter passing.
- 3.4. Library functions.

Specific objectives:

For students to:

- Determine whether a subroutine needs to be an action or a function.
- Send parameters by value and by reference.
- Define formal and actual parameters (arguments).
- Determine whether a formal parameter of an action or function is input, output or input/output.
- Write a program that uses library functions properly.
- Implement and use functions and actions properly.
- Detect and eliminate code repetition.
- Build programs correctly with the help of functions and actions.

Related activities:

- One or more type 2 activities, corresponding to an individual self-learning test.
- One or more type 3 activities, corresponding to an individual self-learning task.
- One phase of activity 4, the project.

Full-or-part-time: 27h

Laboratory classes: 10h

Self study : 17h

TOPIC 4: BASIC ALGORITHMIC TECHNIQUES

Description:

- 4.1. Sequences.
- 4.2. Traversal techniques.
- 4.3. Search techniques.

Specific objectives:

For students to:

- Describe the concept of a sequence.
- Define the sequences associated with a problem.
- Given a sequence problem, determine whether the scheme can be solved by a search technique or a traversal technique.
- Correctly apply traversal and search algorithms.

Related activities:

- A type 1 activity, corresponding to an individual continuous assessment test during laboratory sessions.
- One or more type 2 activities, corresponding to an individual self-learning test.
- One or more type 3 activities, corresponding to an individual self-learning task.
- One phase of activity 4, the project.

Full-or-part-time: 26h

Laboratory classes: 10h

Self study : 16h

TOPIC 5: STRUCTURED TYPES

Description:

- 5.1. Tuples.
- 5.2. Arrays.
- 5.3. Traversal and search techniques in arrays.
- 5.4. Sorting and search algorithms.

Specific objectives:

For students to:

- Write the declaration of a tuple and an array.
- Correctly declare tuple and array variables and access the declarations properly.
- Correctly use pass-by-value and pass-by-reference evaluation with tuples and arrays.
- Generate partially filled arrays.
- Write code that inserts and removes items in a variable-length array.
- Understand and correctly use some sorting algorithms.

Related activities:

- One or more type 2 activities, corresponding to an individual self-learning test.
- One or more type 3 activities, corresponding to an individual self-learning task.
- One phase of activity 4, the project.

Full-or-part-time: 31h

Laboratory classes: 14h

Self study : 17h

TOPIC 6: TOP-DOWN DESIGN

Description:

- 6.1. Top-down design of data structures.
- 6.2. Top-down design of processes.

Specific objectives:

For students to:

- Design an efficient data structure for a given problem.
- Correctly access complex data structures.
- Given a complex problem, carry out top-down design using subroutines.

Related activities:

- Completion of activity 4, the project.

Full-or-part-time: 27h

Laboratory classes: 14h

Self study : 13h

ACTIVITIES

ACTIVITY 1: LABORATORY CONTROLS

Description:

Individual completion in class of an exercise covering all the topic's specific learning objectives. Grading by the teaching staff.

Delivery:

Resolution of the exercise by the student. The activities of type 1 represent 20% of the laboratory assessment.

Full-or-part-time: 3h

Laboratory classes: 3h

ACTIVITY 2: INDIVIDUAL TESTS (CONTINUOUS ASSESSMENT IN ATENEA)

Description:

Individual completion in class of an Atenea quiz covering all or part of the topic's specific learning objectives. Automatic grading.

Delivery:

Student completion of the quiz.

Full-or-part-time: 2h

Self study: 2h

ACTIVITY 3: INDIVIDUAL TASKS (CONTINUOUS ASSESSMENT)

Description:

Individual completion in class of an exercise covering all or part of the topic's specific learning objectives. Grading by the teaching staff.

Delivery:

Completion of the exercise by the student.

Full-or-part-time: 3h

Self study: 3h

ACTIVITY 4: PROJECT

Description:

Group completion of a project covering the main objectives of the course. There will be individual submissions and a final group submission. Grading by the teaching staff.

Delivery:

Students will be required to complete two types of submissions: individual and group submissions.

Full-or-part-time: 20h

Laboratory classes: 5h

Self study: 15h

ACTIVITY 5: CONTENT PRESENTATION AND PRACTICAL WORK

Description:

Individual or group completion in class of exercises related to the topic covered in the session. Group correction by the teaching staff.

Delivery:

No submissions required.

Full-or-part-time: 111h

Laboratory classes: 47h

Self study: 64h

ACTIVITY 6: MIDTERM AND FINAL EXAMS

Description:

In-class written exams (mid-term and final). Graded by the instructor.

Specific objectives:

To evaluate acquired knowledge.

Full-or-part-time: 11h

Laboratory classes: 5h

Self study: 6h

GRADING SYSTEM

Any academic fraud, plagiarism or use or possession of unauthorised means in an assessment activity leads to a mark of zero (0) for the affected test or assignment. Furthermore, in accordance with the University's regulations, if disciplinary proceedings are initiated the subject is recorded as "Pending Assessment" until a decision is reached. These incidents are managed in accordance with the [Action Framework for Academic Integrity in Assessment at the UPC](#).

- Midterm exam: 25%
- Final exam: 35%
- Control sessions: 20%
- Problem solving: 10%
- Final Project: 10%

The assessment of the transversal competence is integrated into the project assessment.

EXAMINATION RULES.

To pass the course, students must attain 50% of the overall grading.

If any of the laboratory assessments is not completed, it will be considered as not graded.

Under no circumstances may any kind of documentation be used in the midterm or final exams.

BIBLIOGRAPHY

Basic:

- Marco, Jordi; Martín, Ángela; Molinero, Xavier; Vázquez, Pere-Pau; Xhafa, Fatos. Programación en C++ para ingenieros. Madrid: Thomson, 2006. ISBN 8497324854.
- Gatus, M.; Lopez, J.; Martin, A.; Romero, O.; Valverde, A.; Xhafa, F.; Esquerra, I.; Amirian, G. Programació pràctica en C++ [on line]. Barcelona: Edicions UPC, 2010 [Consultation: 06/05/2020]. Available on: <http://hdl.handle.net/2099.3/36843>. ISBN 9788498804034.

Complementary:

- Franch, X.; Marco, J.; Molinero, X.; Petit, J.; Xhafa, F. Fonaments de programació: problemes resolts en C++. Barcelona: UPC, 2006. ISBN 9788483018828.
- Marco, J.; Xhafa, F.; Vázquez, P. P. Fonaments d'Informàtica: pràctiques de laboratori. 2a ed. Barcelona: Edicions UPC, 2008. ISBN 9788483019689.

RESOURCES

Hyperlink:

- Atenea. Resource
- Pildoras de C++. <http://www.minidosis.org/#/cursos/FI>- Jutge. Resource <https://www.jutge.org/>- Programació Pràctica en C++.