

Course guide

320002 - FI - Foundations of Computing

Last modified: 11/04/2025

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: 2025 **ECTS Credits:** 6.0 **Languages:** Catalan, Spanish

LECTURER

Coordinating lecturer: Mugica Alvarez, Francisco José
Arratia Quesada, Argimiro Alejandro

Others: Mugica Alvarez, Francisco José
Arratia Quesada, Argimiro Alejandro
Vellido Alcacena, Alfredo
Martín Prat, Ángela
López López, María José
Mylonakis Pascual, Nicolas Eduardo
López Herrera, Josefina
Xhafa Xhafa, Fatos
König, Caroline
Fernández Durán, Pablo
Morrill, Glyn Verden
Amirian, Gerard
Marco Gómez, Jordi
Vázquez Salceda, Javier

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

- Face-to-face lecture sessions

Lectures are given using digital presentations. The presentations will be made available to students on the virtual campus before classes begin to help them follow them. The assessment will be based on mid-semester examinations (or an optional final examination for students who fail the first one).

- Face-to-face practical work sessions

During practical work sessions, students work individually or in small groups of 2-3 on problems and questions under the lecturer's supervision. A collection of problems will be made available on the virtual campus. Systems for self-assessment (with assessment criteria or rubrics), co-assessment (among students) and delivery of reports, corrected by the teacher and returned, are made available to facilitate independent learning.

- Laboratory work sessions

Students work in pairs during laboratory sessions. Guidelines for practicals will be available to students on the virtual campus at the start of the course. Students must hand in a report for each practical. Marks will be based on the work carried out in the laboratory and the reports handed in.

LEARNING OBJECTIVES OF THE SUBJECT

.

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.

Related activities:

- Activity 5, which corresponds to a group research and development project on an introductory topic.

Full-or-part-time: 11h

Laboratory classes: 2h

Guided activities: 1h

Self study : 8h

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:

- One type-1 activity: Individual continuous-assessment test spanning the various laboratory-group sessions.
- One or more type-2 activities: Individual take-home self-directed learning test.
- One or more type-3 activities: Individual take-home self-directed 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 type-1 activity: Individual continuous-assessment test spanning the various laboratory-group sessions.
- One or more type-2 activities: Individual take-home self-directed learning test.
- One or more type-3 activities: Individual take-home self-directed learning task.
- One phase of activity 4, the project.

Full-or-part-time: 27h

Laboratory classes: 10h

Guided activities: 2h

Self study : 15h

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:

- One type-1 activity: Individual continuous-assessment test spanning the various laboratory-group sessions.
- One or more type-2 activities: Individual take-home self-directed learning test.
- One or more type-3 activities: Individual take-home self-directed 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. Tables.
- 5.3. Traversal and search techniques in tables.
- 5.4. Sorting and search algorithms.

Specific objectives:

For students to:

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

Related activities:

- One type-1 activity: Individual continuous-assessment test spanning the various laboratory-group sessions.
- One or more type-2 activities: Individual take-home self-directed learning test.
- One or more type-3 activities: Individual take-home self-directed learning task.
- One phase of activity 4, the project.

Full-or-part-time: 31h

Laboratory classes: 14h

Guided activities: 2h

Self study : 15h

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

Guided activities: 1h

Self study : 12h



ACTIVITIES

ACTIVITY 1: LABORATORY CONTROLS

Description:

Independent work in the classroom to cover all the specific learning objectives of the subject. Professors corrections.

Specific objectives:

At the end of the activity, the student must have achieved specific objectives of the subject.

Material:

Topic notes available (PowerPoint).

Activity statement and the official correction criteria (rubric) available.

Delivery:

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

Full-or-part-time: 2h

Laboratory classes: 2h

ACTIVITY 2: INDIVIDUAL TESTS (CONTINUOUS ASSESSMENT IN ATENEA)

Full-or-part-time: 2h

Self study: 2h

ACTIVITY 3: INDIVIDUAL TASKS (CONTINUOUS ASSESSMENT IN ATENEA)

Full-or-part-time: 3h

Self study: 3h

GRADING SYSTEM

- First examination: 20%
- Second examination: 30%
- Control sessions: 20%
- Application/practicals: 10%
- Final Project: 20%

As part of the evaluation of the project it is included here the evaluation of the common skills

EXAMINATION RULES.

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

BIBLIOGRAPHY

Basic:

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

RESOURCES

Hyperlink:

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