



Course guide

220085 - INF - Fundamentals of Programming

Last modified: 19/04/2023

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

Degree: BACHELOR'S DEGREE IN INDUSTRIAL TECHNOLOGY ENGINEERING (Syllabus 2010). (Compulsory subject).

Academic year: 2023 **ECTS Credits:** 6.0 **Languages:** Catalan, Spanish

LECTURER

Coordinating lecturer: JOSEFINA LÓPEZ HERRERA
MARTA GATIUS VILA

Others: JOSEFINA LOPEZ HERRERA - MARTA GATIUS VILA - FATOS XHAFA XHAFA- ALFREDO VELLIDO
- ANGELA MARTIN PRAT- MARIA JOSE LOPEZ LOPEZ- ARGIMIRO ARRATIA- FRANCISCO
MUGICA- NIKOS MYLONAKIS -GERARD AMIRIAN- CAROLINE KONIG- PABLO FERNANDEZ-
GLYN MORRIL - JORDI MARCO - JAVIER VAZQUEZ

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:

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

Transversal:

CT03 N1. Efficient oral and written communication - Level 1. Planning oral communication, answering questions properly and writing straightforward texts that are spelt correctly and are grammatically coherent.

TEACHING METHODOLOGY

- Theory sessions
- Laboratory sessions
- Tasks and self-study exercises

The sessions consist of an exhibition explanation by the teacher of the contents of the content of the subject: theoretical basis, concepts, methods and results, illustrating them with examples appropriate to facilitate their understanding.

The working sessions will be practical in the classroom of two types:

- sessions in which the teacher guide students in analyzing data and solving problems by applying techniques concepts and theoretical results.
- Sessions Exam

The student will also be studied independently in order to assimilate the concepts and solving exercises (in on paper or computer). This student's work will be supported by online tools.



LEARNING OBJECTIVES OF THE SUBJECT

The main goal of the subject Computer Science is to learn to develop programs using a high level language. In order to pass the subject the student has to be able to:

- Learn the basic concept related to software and hardware: computers structure and operative systems.
- Learn the basic concepts for programming.
- Develop the ability to use basic tools and basic techniques of programming: algorithms and programs.
- Learn to design programs correctly: well-structured, efficient and readable.
- Learn to design data structures representing the data involved in a given problem.
- Design and develop a medium size program.
- Develop their ability to use abstraction in programming patterns to solve real 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

1: INTRODUCTION TO COMPUTERS

Description:

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

Specific objectives:

- 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

Self study : 9h



2. Basic programming concepts

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:

- 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



3. Subprogramms: 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

Self study : 17h

4. BASIC ALGORITHMS

Description:

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

Specific objectives:

Related activities:

- One type-1 activity: Individual continuousassessment 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.

Related activities:

- One type-1 activity: Individual continuousassessment 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



5. Structured type

Description:

- 5.1. Tuples.
- 5.2. Tables.
- 5.3. Traversal and search techniques in tables.
- 5.4. Sorting and search algorithms.

Specific objectives:

- 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

Self study : 17h

6. Design descending

Description:

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

Specific objectives:

- 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:

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: 38h

Laboratory classes: 18h

Self study: 20h

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

Activity 4; midterm exam

Full-or-part-time: 43h

Laboratory classes: 18h

Self study: 25h

Activity 5: Final exam

Full-or-part-time: 45h

Laboratory classes: 20h

Self study: 25h

Activity 6: Project

Full-or-part-time: 19h

Laboratory classes: 4h

Self study: 15h



GRADING SYSTEM

- First examination: 20%
- Second examination: 30%
- Control laboratory 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.

- Any of the control laboratory not performed, will be considered as non-marked
- In no case can any documentation or digital support be used in the partial or final test.

BIBLIOGRAPHY

Basic:

- Gatus, Marta [et al.]. Programació pràctica en C++ [on line]. PRIMERA. Barcelona: Edicions UPC, 2010 [Consultation: 19/05/2020]. Available on: <http://hdl.handle.net/2099.3/36843>. ISBN 9788498804010.

Complementary:

- Marco, J.; Xhafa, F.; Vázquez, P. P. Fonaments d'informàtica: pràctiques de laboratori. 2a ed. Barcelona: Edicions UPC, 2008. ISBN 9788483019689.
- Xhafa, Fatos [et al.]. Programación en C++ para ingenieros. 2006. Madrid: Thomson, 2006. ISBN 8497324854.
- Kernighan, Brian W. El lenguaje de programación C. 2a ed. México: Prentice-Hall Hispanoamericana, 1991. ISBN 9688802050.

RESOURCES

Other resources:

- Web page of the course in atenea
- Divulgació de la programació:
<http://cartesius.upc.es/adminmat/programacio/desenvolupa>
- Pildoras de C++: <http://www.minidosis.org/#/cursos/FI> />- Jutge: <https://www.jutge.org/>