



Course guide

320095 - FI - Foundations of Computing

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 AUDIOVISUAL SYSTEMS ENGINEERING (Syllabus 2009). (Compulsory subject).

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

LECTURER

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

Others: Arratia Quesada, Argimiro Alejandro
Mugica Alvarez, Francisco José
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:

CE02-ESAUD. Basic knowledge of computer use and programming, operating systems, databases, and software applications with engineering applications. (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.

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.

TEACHING METHODOLOGY

- Face-to-face lecture sessions.
- Face-to-face practical work sessions.
- Independent learning and exercises.
- Preparation and completion of group activities subject to assessment.

In the face-to-face lecture sessions, the lecturer will introduce the basic theory, concepts, methods and results for the subject and use examples to facilitate students' understanding.

Practical class work will be covered in three types of sessions:

- a) Sessions in which the lecturer will provide students with guidelines to analyse data for solving problems by applying methods, concepts and theoretical results.
- d) Sessions in which students give presentations of group work.
- c) Examination sessions

Students will be expected to study in their own time so that they are familiar with concepts and are able to solve the exercises set, either manually or with the help of a computer. Students will have access to online tools during independent learning. Students will carry out group work which will be publicly presented during application sessions.

LEARNING OBJECTIVES OF THE SUBJECT

The main aim of the Fundamentals of Informatics course is to teach students how to program using a high-level language. To pass the course, students must be able to:

- Understand the basic concepts associated with computer hardware and software, i.e. the structure of computers and operating systems.
- Understand the fundamental concepts of computer programming.
- Develop the ability to use basic programming tools and techniques: algorithms and programs.
- Design software that is well structured, efficient and readable.
- Design data structures to represent data in a given problem.
- Carry out a medium-scale industrial programming project.
- Develop the capacity for abstraction in the use of programming patterns for solving real problems.

STUDY LOAD

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

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: 7h

Laboratory classes: 2h

Self study : 5h



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 ALGORITHM

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.
- Determine, given a sequence problem, 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:

- Completion of activity 4, the project.

Related activities:

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.

Full-or-part-time: 31h

Laboratory classes: 14h

Guided activities: 2h

Self study : 15h



ACTIVITIES

PROJECT

Description:

Cooperative work aimed to work with very specific aspects of the subject (activity type 4). There will be 3 sessions with 3 groups of 3 people to work items 3, 5 and 6. The activity is based on collaborative work with experts and each group at the end of each session will implement a small program about the topic.

Specific objectives:

At the end of the activity, the student must have achieved all the objectives of the course.

Material:

Examples of projects solved.

Delivery:

The program is conducted at the end of the session.
The evaluation of this project activity included within the Type 4.

DIRECTED ACTIVITIES TEST

Description:

Cooperative work based on the sharing of a basic computer introductory unit. It will work with 3 groups of 3 components should discuss the approach to the topic and relevance of the data sources and shall make proposals for improvement. It will also work on the best way to introduce the topic publicly.

Specific objectives:

After the activity of 5 (including this one), a student must have achieved all the specific objectives of the subject 1. In activity 5 (which includes this) generic skills of information retrieval and presentation of oral work

Material:

Wording and documentation related to the information seeking skills and oral presentation.

Delivery:

Wording and documentation related to the information seeking skills and oral presentation.

LABORATORY CONTROL

GRADING SYSTEM

Mid-semester examination: 20%

Final examination: 30%

Quizzes: 20%

Problem solving: 10%

Project: 20%

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

For those students who meet the requirements and submit to the reevaluation examination, the grade of the reevaluation exam will replace the grades of all the on-site written evaluation acts (tests, midterm and final exams) and the grades obtained during the course for lab practices, works, projects and presentations will be kept.

If the final grade after reevaluation is lower than 5.0, it will replace the initial one only if it is higher. If the final grade after reevaluation is greater or equal to 5.0, the final grade of the subject will be pass 5.0.



EXAMINATION RULES.

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

BIBLIOGRAPHY

Basic:

- Marco, J.; Martín, Á.; Molinero, X.; Vázquez, P-P.; Xhafa, F. Programación en C++ para ingenieros. Madrid: Thomson-Paraninfo, 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.