

## Course guide

### 300012 - IO - Introduction to Computers

**Last modified:** 01/06/2023

<b>Unit in charge:</b>	Castelldefels School of Telecommunications and Aerospace Engineering	
<b>Teaching unit:</b>	701 - DAC - Department of Computer Architecture.	
<b>Degree:</b>	BACHELOR'S DEGREE IN NETWORK ENGINEERING (Syllabus 2009). (Compulsory subject). BACHELOR'S DEGREE IN TELECOMMUNICATIONS SYSTEMS ENGINEERING (Syllabus 2009). (Compulsory subject).	
<b>Academic year:</b> 2023	<b>ECTS Credits:</b> 6.0	<b>Languages:</b> Catalan, Spanish

#### LECTURER

---

**Coordinating lecturer:** Definit a la infoweb de l'assignatura.

**Others:** Definit a la infoweb de l'assignatura.

#### PRIOR SKILLS

---

No prior knowledge is required.

#### DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

---

**Specific:**

1. CE 2 TELECOM. Students will acquire basic knowledge of the use and programming of computers, operating systems, databases and computer programs used in engineering. (CIN/352/2009, BOE 20.2.2009)

**Generical:**

5. EFFICIENT USE OF EQUIPMENT AND INSTRUMENTS - Level 1: Using instruments, equipment and software from the laboratories of general or basic use. Realising experiments and proposed practices and analyzing obtained results.

**Transversal:**

2. SELF-DIRECTED LEARNING - Level 1. Completing set tasks within established deadlines. Working with recommended information sources according to the guidelines set by lecturers.
3. 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.
4. TEAMWORK - Level 1. Working in a team and making positive contributions once the aims and group and individual responsibilities have been defined. Reaching joint decisions on the strategy to be followed.

#### TEACHING METHODOLOGY

---

It is highly recommended that students have a personal computer (ideally a notebook) with an Internet connection. We offer a borrowing service for netbooks.

This course combines the following teaching methods:

- Independent learning, as students will work on self-study material at home.
- Cooperative learning, as students will organise themselves in small groups to perform tasks.
- Project-based learning, as students will develop a team project during the second part of the subject.
- Self-assessment and co-assessment of assignments.

## LEARNING OBJECTIVES OF THE SUBJECT

On completing the Introduction to Computers course, students will be able to:

- Describe the elements that make up the Von Newmann architecture and their functions.
- Describe the relationship between the elements of computer architecture and the structures of high-level languages.
- Describe the usual functions of an operating system and give examples of the functions that an operating system offers to the programmer and the user.
- Develop computer programs in the C# language using basic data and control structures, traverse and search algorithms, text files and functions.
- Use the Microsoft Visual Studio programming environment properly.
- Present their work appropriately, in both oral and written form.
- Learn independently: assimilate information from references, search for information relevant to the learning process and identify their errors.
- Work in a group: identify the tasks to perform, share these tasks and integrate the results, resolve conflicts occurring within the group and identify areas for improvement.

## STUDY LOAD

Type	Hours	Percentage
Self study	84,0	56.00
Guided activities	23,0	15.33
Hours small group	43,0	28.67

**Total learning time:** 150 h

## CONTENTS

### Computer Architecture and Operating Systems

#### Description:

- 1.1 Basic elements of the Von Newmann architecture
- 1.2 Assembly language
- 1.3 Correspondence between assembly languages and high-level languages
- 1.4 Functions and services of operating systems

#### Related activities:

Activity 1: Introduction to Computer Architecture and Operating Systems: The Simple Machine

**Full-or-part-time:** 34h 30m

Laboratory classes: 9h 54m

Guided activities: 5h 18m

Self study : 19h 18m

### Basic C# Programming

**Description:**

- 2.1 Basic data types
- 2.2 Conditional statements and loops
- 2.3 One-dimensional and multidimensional arrays
- 2.4 Basic algorithm types

**Related activities:**

Activity 2: Introduction to C# Programming

**Full-or-part-time:** 34h 30m

Laboratory classes: 9h 54m

Guided activities: 5h 18m

Self study : 19h 18m

### Advanced C# Programming

**Description:**

- 2.5 Structures
- 2.6 Files
- 2.7 Functions

**Related activities:**

Activity 3: Programming Project

**Full-or-part-time:** 46h 30m

Laboratory classes: 13h 18m

Guided activities: 7h 06m

Self study : 26h 06m

### The Programming Environment

**Description:**

- 4.1 Project design
- 4.2 Compiling and building: error resolution
- 4.3 Runtime error resolution: debuggers

**Related activities:**

Activity 2: Introduction to C# programming

Activity 3: Programming Project

**Full-or-part-time:** 34h 30m

Laboratory classes: 9h 54m

Guided activities: 5h 18m

Self study : 19h 18m



## ACTIVITIES

### INTRODUCTION TO COMPUTER ARCHITECTURE AND OPERATING SYSTEMS: THE SIMPLE MACHINE

**Description:**

The directed activities and independent learning will involve self-study with the support of a simple machine simulator, as well as individual exercises (with frequent self-assessments) and small group exercises. The class sessions will be used to:

- Resolve doubts in weekly assignments, in small groups.
- Resolve the most common doubts with the lecturer.
- Provide some explanatory sessions on key aspects of the material.
- Carry out individual and small group exercises. Thus cooperative work will be a key element of the work methodology.

**Specific objectives:**

On completing this activity, students will be able to:

- Name the elements that make up the Von Newmann architecture and describe their functions.
- Describe the assembly language instructions of the simple machine.
- Simulate the execution of small programs on the simple machine.
- Describe the relationship between the basic structures of high-level languages and the elements of assembly language.

**Material:**

- Self-study material
- Simulator: The Simple Machine
- Individual and group exercises
- Detailed programme of activities and due dates

All of the material is available through Atenea.

**Delivery:**

The activity has a series of individual and group assignments (with at least one due per week). Feedback processes will be connected with these assignments.

Students must hand in at least 80% of the course assignments on time to pass the course.

The assignments test basic knowledge and are subject to assessment.

**Full-or-part-time:** 34h 30m

Laboratory classes: 9h 54m

Guided activities: 5h 18m

Self study: 19h 18m

## INTRODUCTION TO C# PROGRAMMING

### Description:

The directed activities and independent learning will involve self-study with the support of the Microsoft Visual Studio 2008 programming environment, as well as individual exercises (with frequent self-assessments) and small group exercises. The class sessions will be used to:

- Resolve doubts in weekly assignments, in small groups.
- Resolve the most common doubts with the lecturer.
- Provide some explanatory sessions on key aspects of the material
- Carry out individual and small group exercises. Thus cooperative work will be a key element of the work methodology.

### Specific objectives:

On completing this activity, students will be able to:

- Write correct programs using the basic elements of the C# language (conditional statements and loops with basic data types such as integer, string, real and array).
- Choose and apply the correct algorithm type (traverse or search) in order to solve a simple programming problem.
- Carry out the necessary operations in order to create a project, write the code of a project in the Microsoft Visual Studio 2008 programming environment, compile, build and run it, and resolve basic compilation and runtime errors.

### Material:

- Self-study material
- The Microsoft Visual Studio programming environment
- Individual and group exercises
- Detailed programme of activities and due dates.

All of the material is available through Atenea.

### Delivery:

The activity has a series of individual and group assignments (with at least one due per week). Feedback processes will be connected with these assignments.

Students must hand in at least 80% of the course assignments on time to pass the course.

The assignments test basic knowledge and are subject to assessment.

**Full-or-part-time:** 51h 24m

Laboratory classes: 14h 48m

Guided activities: 7h 48m

Self study: 28h 48m

## PROGRAMMING PROJECT

### Description:

Students will develop a programming project in teams of three or four. Project-based learning is used, so they have to independently learn the material they need to achieve the project goals. Specific cooperative learning tasks will be carried out using the Jigsaw technique. The directed activities and independent learning will involve:

- Self-study
- Individual and group exercises
- Individual project tasks
- Group meetings to carry out project tasks

The activities to be carried out in the class sessions are:

- Resolution of doubts in weekly assignments, in small groups
- Resolution of the most common doubts with the lecturer
- Explanatory sessions on key aspects of the material
- Individual and group exercises
- Individual project tasks
- Group meetings to carry out project tasks

In this activity, special attention will be paid to the written and oral presentation of the work to be carried out by the teams.

### Specific objectives:

On completing this activity, students will be able to:

- Properly choose and define the data structures necessary in order to solve a simple programming problem.
- Define functions and use them properly.
- Write programs that read and generate text files.

In addition, in the context of this activity the students will develop the generic competencies outlined above.

### Material:

- Self-study material
- The Microsoft Visual Studio programming environment
- Requirements for individual and group exercises
- Detailed programme of activities and due dates.

All of the material will be available through Atenea.

### Delivery:

The activity has a series of individual and group assignments (with at least one due per week). Feedback processes will be connected with these assignments.

Students must hand in at least 80% of the course assignments on time in order to pass the course.

The assignments test basic knowledge and are subject to assessment.

The activity also has three large assignments related to the project results and these will contribute a total of 40% to the final mark for the course.

**Full-or-part-time:** 64h 06m

Laboratory classes: 18h 18m

Guided activities: 9h 54m

Self study: 35h 54m

## GRADING SYSTEM

The assessment weighting for the course is as follows:

- Assignments (20%)
- Examinations (30%)
- Project (40%)
- Attitude and participation (10%)

## EXAMINATION RULES.

In order to pass the course, students must hand in at least 80% of the assignments for the course on time. It will also be necessary for students to pass all of the basic knowledge tests (or all except one).



## BIBLIOGRAPHY

---

### Basic:

- Sharp, John; Jagger, Jon. Microsoft Visual C#. NET : step by step [on line]. Redmond: Microsoft Press, 2003 [Consultation: 22/11/2023]. Available on : <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pq-origsite=primo&docID=7117342>. ISBN 0735619093.
- Platt, David S. Introducing Microsoft .NET. 3rd ed. Redmond: Microsoft Press, 2003. ISBN 0735619182.

### Complementary:

- Ceballos Sierra, Francisco Javier. El Lenguaje de programación C#. Madrid: RA-MA, 2002. ISBN 8478975004.
- Archer, Tom; Whitechapel, Andrew. Inside C#. 2nd ed. Redmond, Washington: Microsoft Press, 2002. ISBN 0735616485.