

Degree programme	Type	Course
Business and Information Technology	FB	1

Contact lecturer

Name : Roger Malet Munté

Email : roger.malet@uab.cat

Prerequisites

To be able to correctly follow this subject, some previous algorithmic knowledge and problem solving are required. This knowledge is achieved through the subject "Introduction to Problem Solving and Algorithm Design" (102151), a first year, first semester course which gives the necessary mechanisms to deal with the problems that must be resolved through programming in a high level language.

Group languages

You can consult this information at the [end](#) of the document.

Objectives

The goals of this subject are to introduce the basic concepts of the programming of computer applications to solve problems that can occur in the business field. The idea of this course is to be able to implement algorithms in a programming language in order to be able to generate final solutions that solve specific problems and automate certain computer processes that occur in the companies. In addition, understanding the process of generating applications must allow a greater understanding of the more complex computer processes that occur in this same field.

Thus, the training objectives of the subject can be summarized in:

- Understand and use the basic structures of structured programming
- Implement algorithms in a high-level programming language

Learning outcomes

- CM28 (Use basic structured programming structures to solve business problems.) Use basic structured programming structures to solve business problems.
- KM14 (Identify algorithm and program analysis techniques.) Identify algorithm and program analysis techniques.
- SM26 (Coding algorithms using structured programming techniques.) Coding algorithms using structured programming techniques.

Contents

Topic 1: Introduction

- Goal:

Introduce the way computers process information, the role of programming languages and the process of developing applications.

- Contents:

1.1. Computers and algorithms: Information processing

1.2. Programming languages and programs

1.3. Software development with the computer

1.4. Storing information on the computer

Topic 2: Introduction to C# programming

- Goal:

First contact with a programming language, and its main features.

- Contents:

2.1. My first program

2.2. Elements of the programming language

2.3. The data in the programming language

2.4. The variables

2.5. Working with data

2.6. Input/output

2.7. Examples

Topic 3: Control structures

- Goal:

Introduction to structures that allow a program to make decisions and apply basic algorithms.

- Contents:

3.1. General structure of a program: control structures

3.2. Structures sequence

3.3. Selection structures

3.4. Iteration structures

3.5. More about the increment / decrement operator

3.6. Examples

Topic 4: Structured programming

- Goal:

Introduction to the structures that allow to organize the code so that the applications are readable and scalable.

- Contents:

4.1. Structured and modular programming

4.2. Procedures and functions

4.3. Functions / Procedures

4.4. Variables and i parameters

4.5. Recursiveness

4.6. Examples

Topic 5: Vectors (arrays): Lists and tables

- Goal:

Study of the elements that allow to create lists and tables of elements in C#.

- Contents:

5.1. Introduction

5.2. Vectors (lists)

5.3. Chains (lists) of characters

5.4. Matrices

5.5. Parameter passing

5.6. Operating with vectors (lists)

5.7. Examples

Topic 6: Structures

- Goal:

Introduction to the types of data defined by the programmer, which help to create more complex types than those defined by the C# language itself.

- Contents:

6.1. Introduction to structures (registries)

6.2. Nested structures

6.3. Structures passing

6.4. Examples

Topic 7: files

- Goal:

Introduction to the mechanisms to work with files in C#.

- Contents:

7.1. Introduction

7.2. Writing files

7.3. Reading files

7.4. Read/write functions

7.5. Random access files

7.6. Parameter passing on the argument

Topic 8: Object Oriented (O.O.) programming introduction

- Goal:

Introduction to the programming paradigm that sees a program as a series of objects or entities that are related to each other.

- Contents:

8.1. What is O.O. programming?

8.2. O.O. programming characteristics.

8.3. Class and instance.

8.4. Examples

Learning activities and methodology

Title	Hours	ECTS	Learning outcomes
Laboratory sessions	10	0.4	CM28, KM14, SM26
Personal tutorials	15	0.6	
Theoretical classes	29.5	1.18	CM28, KM14
Preparation and resolution of exercises and practices	81.5	3.26	CM28, KM14, SM26
Problems classes	10	0.4	KM14, SM26

The teaching methodology will be based on three types of activity:

- Guided activity: theoretical classes, practices, and problem analysis
- Supervised activity: attendance to tutorials and execution of executions with scheduled follow-up
- Autonomous activity: student study and case resolution, individually or in groups

Annotation: within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Assessment

Continuous assessment activities

Title	Weight	Hours	ECTS	Learning outcomes
Class exercises	10	0	0	CM28, KM14, SM26
Partial exam	25%	2	0.08	CM28, KM14, SM26
Delivery of practices	30	0	0	KM14, SM26
Final exam	35%	2	0.08	CM28, KM14, SM26

This subject does not provide for a single assessment system.

The course will be evaluated continuously, through exams, problems related to theory classes and specific work from the laboratory sessions. The weight of each of the components will be:

- 30% for the average of the practical training marks,
- 10% of the exercises proposed in class individually or in groups,
- 25% partial exam, which will be carried out around the middle of the semester on the date set by the Faculty, and,
- 35% final exam, which will be held at the end of the semester, on the date set by the Faculty.

The subject will be passed if a minimum grade of 5.0 is achieved in the weighted average of the grades obtained in the activities described above. To make the average, the following conditions must be met:

- It will be necessary to have submitted all the practices, and to have obtained a minimum grade of 5.0.
- The weighted average of the exams must exceed 3.5.

Students who have obtained a mark higher than 3.5 but who do not reach 5 will have the right to a recovery, in the terms described below (retake process).

A student will be considered "non-evaluable" in the subject when he or she has not participated in any of the assessment activities.

Calendar of evaluation activities

The dates of the evaluation activities (exercises, assignments ...) will be announced well in advance during the semester.

The dates of the midterm and final exams are scheduled in the assessment calendar of the Faculty.

"The dates of evaluation activities cannot be modified, unless there is an exceptional and duly justified reason why an evaluation activity cannot be carried out. In this case, the degree coordinator will contact both the

teaching staff and the affected student, and a new date will be scheduled within the same academic period to make up for the missed evaluation activity." **Section 1 of Article 115. Calendar of evaluation activities (Academic Regulations UAB).** Students of the Faculty of Economics and Business, who in accordance with the previous paragraph need to change an evaluation activity date must process the request by filling out an Application for exams' reschedule at https://eformularis.uab.cat/group/deganat_feie/application-for-exams-reschedule

Grade revision process

After all grading activities have ended, students will be informed of the date and way in which the course grades will be published. Students will be also be informed of the procedure, place, date and time of grade revision following University regulations.

Retake Process

"To be eligible to participate in the retake process, it is required for students to have been previously been evaluated for at least two thirds of the total evaluation activities of the subject." **Section 3 of Article 112 ter. The recovery (UAB Academic Regulations).** Additionally, it is required that the student to have achieved an average grade of the subject between 3.5 and 4.9.

The date of the retake exam is posted in the calendar of evaluation activities of the Faculty. Students taking this exam and passing will get a grade of 5 for the subject. For the students that do not pass the retake, the grade will remain unchanged, and hence, will fail the course.

Irregularities in evaluation activities

The completion of assessment activities is subject to the provisions set out in this course guide and in the "Policy of the School of Economics and Business on the Detection of Irregularities during Assessment Activities", which regulates the conditions under which assessment tasks are conducted and the procedures applicable in cases where indications of irregularities are detected. Students are encouraged to consult the policy.

Despite other disciplinary measures deemed appropriate, and in accordance with current academic regulations, *"whenever a student makes any irregularity that could lead to a significant variation in the grade of an evaluation activity, it will be graded with a 0, regardless of the disciplinary process that can be instructed. In case of occurrence of various irregularities in the evaluation of the same subject, the final grade of this subject will be 0"*. **Section 10 of Article 116. Results of the evaluation. (UAB Academic Regulations).**

Final note: The proposed evaluation activities may undergo some changes according to the restrictions imposed by the health authorities on on-campus courses.

Bibliography

[El tutorial de Python - documentación de Python - 3.13.5](#)

[Python Tutorial - Tutorialspoint](#)

[A Beginners Guide To Python 3 Programming by John Hunt \(zlib.pub\)](#)

Software

- Python 3 ([Welcome to Python.org](https://www.python.org/))
- Visual Studio Code ([Visual Studio Code - Code Editing. Redefined](https://code.visualstudio.com/))

Course groups and languages

The information provided is provisional until November 30. After this date, you will be able to consult the language of each group through this [link](#). To access the information, you will need to enter the course CODE

Type of teaching	Group	Language	Semester	Shift
(TE) Theory	20	Catalan	second semester	morning-mixed
(TE) Theory	21	Catalan	second semester	morning-mixed
(PAUL) Classroom practices	201	Catalan	second semester	morning-mixed
(PLAB) Practical laboratories	201	Catalan	second semester	morning-mixed
(PAUL) Classroom practices	202	Catalan	second semester	morning-mixed
(PLAB) Practical laboratories	202	Catalan	second semester	morning-mixed
(PLAB) Practical laboratories	203	Catalan	second semester	morning-mixed
(PLAB) Practical laboratories	204	Catalan	second semester	morning-mixed