

Degree programme	Type	Course
Mathematics	OP	4

Contact lecturer

Name : Vicenç Soler Ruíz

Email : vicenc.soler@uab.cat

Group languages

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

Prerequisites

It is convenient to have learned the knowledge in basic programming in C in the first years of the Degree or in other subjects, although it is not essential.

Objectives

The subject has two different parts. A first objective is to achieve a good level of programming in C, so that students will be able to structure an application and to program a complex algorithm. A second objective is to achieve a good training in object-oriented programming (mainly Python, but also using Java or C++), creation of libraries and to serve as a basis for learning any other object-oriented language.

The practical exercises to turn in are based on implementing two games: one in C and the other in Java or Python.

The development of games, as a basis for the practices of the subject, provides the obligation to use many different aspects of the programming languages and makes the practical exercises more attractive for the student. All this motivates the student to be more interested in the development of the subject.

Learning outcomes

1. Effectively use bibliographies and electronic resources to obtain information.
2. Students must have and understand knowledge of an area of study built on the basis of general secondary education, and while it relies on some advanced textbooks it also includes some aspects coming from the forefront of its field of study.
3. Students must be capable of applying their knowledge to their work or vocation in a professional way and they should have building arguments and problem resolution skills within their area of study.
4. Develop critical thinking and reasoning and know how to communicate it effectively, both in one's own languages and in a third language.
5. Students must be capable of communicating information, ideas, problems and solutions to both specialised and non-specialised audiences.
6. Students must develop the necessary learning skills to undertake further training with a high degree of autonomy.
7. Generate innovative and competitive proposals for research and professional activities.
8. Actively demonstrate high concern for quality when defending or presenting the conclusions of one's work.

9. Work in teams
10. Be able to fully or partially implement the program product and be able to design the verification and validation methods.
11. Know the different activities implied in the phases of the life-cycle of program products and their applications.
12. Understand the architecture and specifications for operating program products, understanding their nature and the possibilities offered by different programming languages.
13. Know how to provide scientifically valid and technologically up-to-date solutions.
14. Know how to analyse, design and implement computer-based systems in general, using methods and techniques that guarantee their efficacy and efficiency.
15. Master the methodologies and tools for developing information systems, database management systems and tools for automating the development of program products.
16. Have an understanding of communication machinery and systems.

Contents

First part.

1. Reminder of structured and modular programming applied to C.

1.1. Basic reminder

1.1.1. Structures and data types

1.1.2. Sequence, control and iteration structures

1.1.3. Procedures and functions

1.2. Dynamic structures

1.2.1. Organization of pointers and memory addresses

1.2.2. Basic structures: stacks, lists and queues..

1.2.3. Trees

1.2.4. Dynamic matrices

2. Recursivity.

2.1. Fundamentals of recursivity

2.2. Search algorithms using trees:

2.2.1. Priority amplitude and priority depth.

2.2.2. Backtracking.

Second part

3. Object Oriented Programming (Python and Java/C++)

3.1. General concepts and features.

3.2. Classes and objects: inheritance, overriding, ...

3.3. Object oriented analysis and design.

4. Python and Java/C++

4.1. Languages features.

4.2. Classes, loops, functions and languages structures.

*Unless the requirements enforced by the health authorities demand a prioritization or reduction of these contents.

Learning activities and methodology

Title	Hours	ECTS	Learning outcomes
Personal study and development of practices	97	3.88	10
Lectures of theory, problems and practices	45	1.8	10

Lectures:

The concepts of the subject will be presented. Emphasis will be placed on the interpretation of the results and the relationship between these concepts and their applications. Examples will be presented in order to allow students to autonomously address how to solve problems and the resolution of the proposed problems will be discussed.

The methodology of the subject will be entirely practical. This implies that classes will be taught on a computer. The theoretical concepts will be explained in each session and complemented by exercises proposed by the teacher, which will be coded directly on the computer, under his supervision.

Tutoring:

The teacher will supervise and evaluate the student work. Students will propose the doubts that have arisen in the resolution of the proposed exercises.

Doubts will be resolved and the resolution of the problems will be discussed.

Autonomous activities:

The practical assignments must be developed individually. There is no point, in this subject, to create groups of more than one student to develop the work to be turned in.

*The proposed teaching methodology may experience some modifications depending on the restrictions to face-to-face activities enforced by health authorities.

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
Practices delivery	66%	0	0	1, 2, 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 16
Exam	33%	4	0.16	2, 3, 4, 8, 15
Resit exam	33%	4	0.16	2, 3, 4, 8, 15

Theory and problems module (weight 33%).

The evaluation of this part will be done through 2 exams: midterm and final exams.

To pass the subject, a minimum of 2.5 points must be obtained in the part of the exams. The midterm exam has a weight of 40% and the final 60% of the exams grade.

Practical module (weight 66%)

Two practical exercises will be turned in during the course. Each of these exercises will count for 33% of the final grade.

Exams

The maximum grade (Excellent with Honor) will not be awarded to students who must go to the retake exam.

The grade "Not assessable" will be applied to students who do not take any of the exams.

The retake exam is only applicable to students who have not passed the subject.

Use of AI

The use of AI is not allowed in exams. The exam must be developed individually by each student, without any external help and only the material specified by the teacher will be allowed to be brought.

When developing the practices, the use of AI may be allowed in specific tasks that do not represent the main part of the practice. The student must clearly identify which parts have been generated with this technology and specify the tools used. Its use will not be allowed in more than 10% of the code and the teacher must be consulted if it is allowed to use AI in the part that is intended to be developed. The objective is that the practices must be developed by the students individually and AI can help in a specific task, but cannot be the main part of the practice.

Any assessable activity that includes fragments generated with AI (without being under the previous

conditions) or copied from any student will be considered a lack of academic honesty and may lead to a partial or total penalty in the grade of the activity, or greater sanctions in serious cases.

Single evaluation

Programació Avançada has not single evaluation.

Bibliography

- K. N. King , "C Programming: A Modern Approach", Ed. WW Norton & Co, 2008
- Raúl González Duque, "Python para todos", <http://mundogeek.net/tutorial-python/>
- Python tutorial", <https://www.tutorialspoint.com/python/>
- Mark Lutz, "Learning Python", Ed. O'Reilly, 2013
- "The C++ Programming Language", B.Stroustrup, 3ªed.,Addison-Wesley, 1997.
- P.Naughton, H.Schildt , "Java2: The Complete Reference", Osborne/McGraw-Hill, 2000
- "Core Java" Volumes I y II, C.S.Horstmann, G.Cornell, Sun Microsystems Press.

Software

<https://www.codeblocks.org/downloads/binaries/>

<https://www.eclipse.org/downloads/>

<https://www.python.org/downloads/>

<https://code.visualstudio.com/download>

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
(PLAB) Practical laboratories	1	Catalan	first semester	morning-mixed