UAB
Universitat Autònoma de Barcelona

Programming

Code: 107739
Credits: 6

2026/2027

Degree programme	Type	Course
Interactive Communication	FB	2

Contact lecturer

Name : Maria Daniela Cordova Pintado

Email : daniela.cordova@uab.cat

Group languages

You can consult this information at the end of the document.

Prerequisites

t is recommended that students have taken the course Introduction to Programming and have basic knowledge of algorithmics and Python programming. In particular, students should be familiar with the use of variables, expressions, conditional structures, repetitive structures, lists and the resolution of simple problems through programs.

The course builds on this initial knowledge in order to consolidate and expand it towards a more structured, modular approach to programming applied to interactive communication projects.

Objectives

The main objective of the course is to consolidate the programming foundations previously acquired and to progress towards the development of simple computational solutions applied to the field of interactive communication.

Learning outcomes

- CM09 (Apply the principles of algorithms and programming to proposals for interactive communicative projects within the framework of a professional practice that is sensitive to society's problems and challenges.) Apply the principles of algorithms and programming to proposals for interactive communicative projects within the framework of a professional practice that is sensitive to society's problems and challenges.
- CM10 (Design interactive communicative projects with the Python programming language in accordance with the principles of ethical responsibility and respect for fundamental rights and duties, diversity and democratic values.) Design interactive communicative projects with the Python programming language in accordance with the principles of ethical responsibility and respect for fundamental rights and duties, diversity and democratic values.
- CM11 (Create and develop applications with the Python programming language in the field of interactive communication autonomously with initiative and proactivity.) Create and develop applications with the Python programming language in the field of interactive communication autonomously with initiative and proactivity.
- KM07 (Explain the basic conceptual and practical concepts, languages and application of algorithms and programming in a programming environment.) Explain the basic conceptual and practical concepts, languages and application of algorithms and programming in a programming environment.

- KM09 (Distinguish the most complex programming concepts in Python.) Distinguish the most complex programming concepts in Python.
- SM08 (Use programming principles to turn a specific problem into a programme that solves it.) Use programming principles to turn a specific problem into a programme that solves it.
- SM10 (Use the Python programming language in interactive communication projects) Use the Python programming language in interactive communication projects

Contents

The course explores Python programming in greater depth, building on the knowledge previously acquired in introductory courses. The contents are organized progressively, starting with the consolidation of structured programming and moving towards modular code organization, the use of data structures, object-oriented programming and the development of small projects applied to interactive communication.

1. Data structures and basic information processing
2. Functions and modularity
3. Introduction to object-oriented programming
4. Development of small applied projects

Learning activities and methodology

Title	Hours	ECTS	Learning outcomes
Laboratory practice sessions	15	0.6	CM09, CM10, CM11, KM09, SM08, SM10
Classroom practice sessions	9	0.36	CM09, CM11, KM07, SM08, SM10
Independent work, study and preparation of activities	30	1.2	CM09, CM10, CM11, KM07, KM09, SM08, SM10
Theory classes	20	0.8	CM09, KM07, KM09, SM08
Practice and project supervision	16	0.64	CM10, CM11, SM08, SM10

The course will be developed through a progressive methodology combining theoretical sessions, classroom practice sessions, laboratory practices and students' independent work. The approach starts from the consolidation of previous programming knowledge and progresses towards problem solving, code organization and the development of small projects applied to interactive communication.

The theoretical sessions will introduce the fundamental concepts of each content block and relate them to practical examples.

The classroom practice sessions will allow students to work on guided exercises, discuss possible solutions and reinforce their understanding of the concepts covered.

Laboratory practices will play a central role in the course. In these sessions, students will develop programs in Python, solve practical exercises and progress in the construction of small prototypes.

Independent work will include studying the course materials, preparing and completing exercises, developing practical assignments and preparing assessment activities. The detailed schedule of sessions, practical work and submissions will be published on the course Virtual Campus.

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
Laboratory practices	0.4	25	1	CM09, CM10, CM11, KM09, SM08, SM10
Attendance, follow-up and active participation	0.1	5	0.2	CM09, CM11, KM07, SM08, SM10
Theoretical-practical tests	0.3	10	0.4	CM09, KM07, KM09, SM08
Applied programming project	0.2	20	0.8	CM10, CM11, SM08, SM10

The assessment of the course will be continuous and will combine follow-up activities, theoretical-practical tests, laboratory practices and an applied project. The aim of the assessment is to verify both the understanding of the fundamental concepts of programming and the students' ability to apply them to problem solving and to the development of small programs related to interactive communication.

The assessment activities and their percentages of the final grade will be as follows: attendance, follow-up and active participation, 10%; theoretical-practical tests, 30%; laboratory practices, 40%; and applied programming project, 20%. The sum of these activities corresponds to 100% of the final grade.

The theoretical-practical tests will assess the understanding of fundamental programming concepts, the reading and interpretation of code, and problem solving. The laboratory practices will assess the progressive application of the course contents through exercises and programs developed in Python. The applied project will consist of the design and development of a small program or prototype related to the field of interactive communication.

To pass the course, students must obtain a minimum final grade of 5 out of 10. The specific conditions for each activity, as well as the schedule for submissions and tests, will be published on the Virtual Campus.

Students may take the reassessment only if they have completed enough assessment activities during the course. Specifically, they must have been assessed in activities that, taken together, represent at least two thirds of the final grade for the course. The reassessment will consist of a theoretical-practical recovery test and/or the correction or resubmission of failed practical activities, according to the instructions established by the teaching staff on the Virtual Campus. The maximum grade for reassessed activities will be 5.

This course does not include the single-assessment system.

The use of Artificial Intelligence tools will only be allowed when expressly authorized by the teaching staff. Students must be able to explain, justify and defend the code submitted.

If a student commits any irregularity that may lead to a significant variation in the grade of an assessment activity, that activity will be graded with 0, regardless of any disciplinary proceedings that may be initiated. In the event of several irregularities in the same course, the final grade will be 0.

Bibliography

The specific bibliography for the course will be provided on the Virtual Campus according to the activities, practical sessions and projects developed during the course.

The following general reference resources are proposed:

Python Software Foundation. *Python Documentation*. Available at: https://docs.python.org/3/

Python Software Foundation. *Python*. Available at: https://www.python.org/

Python Institute. *Python Institute learning resources*. Available at: https://pythoninstitute.org/

Downey, A. B. *Think Python: How to Think Like a Computer Scientist*. Green Tea Press. Available at: https://greenteapress.com/wp/think-python-2e/

Software

Python 3

Visual Studio Code

Python extensions for Visual Studio Code

Web browser

Support tools for running, debugging and testing Python programs

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	1	Catalan/Spanish	first semester	afternoon
(PAUL) Classroom practices	1	Catalan/Spanish	first semester	afternoon
(PLAB) Practical laboratories	1	Catalan/Spanish	first semester	afternoon
(PLAB) Practical laboratories	2	Catalan/Spanish	first semester	afternoon