UAB
Universitat Autònoma de Barcelona

Introduction to Programming

Code: 104383
ECTS Credits: 6

2024/2025

Degree	Type	Year
2503740 Computational Mathematics and Data Analytics	FB	1

Contact

Name: Roberto Benavente Vidal

Email: robert.benavente@uab.cat

Teachers

Roberto Benavente Vidal

Teaching groups languages

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

Prerequisites

Given that this is an introductory course, it is assumed that students do not have any prior knowledge of the subject.

However, it is advisable to have a basic knowledge of any platform (windows, mac or linux).

In face-to-face activities in the classroom, it is essential to have a laptop to carry out the assignments of the subject.

Objectives and Contextualisation

This subject has a general and introductory character in the programming. It will be deepened in the study of the methodological aspects of programming and the learning of a high level language. Therefore, the general objectives that are proposed for the subject are the following:

- Understand the software life cycle: analyze the problem (understand what is being asked), design (propose a solution to the problem), implementation (coding in a programming language of the chosen solution), test (completion of a test systematically to ensure the correctness of the implemented solution).
- Provide the student with the ability to design algorithms for solving problems, introducing a rigorous and structured programming methodology progressively and systematically, based essentially on the technique of downstream design of algorithms.

- Introduce the student in a real programming language. It is intended that the student perceives the difference between the flexibility of pseudo-algorithmic notation that is used in the first topics and the strict syntax of a real programming language, both in its lexical aspects (valid words of the language), syntactic (rules to combine) and semantic (meaning of the same).
- To habit the student to develop programs following a style norms that seek to achieve quality programs. These rules of style include those that facilitate the compression of the code, such as the use of comments, indentation of the code, the use of names suitable for data types, etc.

Learning Outcomes

1. CM06 (Competence) Develop effective algorithmic solutions to computational problems in accordance with the established requirements.
2. CM06 (Competence) Develop effective algorithmic solutions to computational problems in accordance with the established requirements.
3. CM07 (Competence) Analyse the computational complexity of the algorithmic solutions to develop and implement the one that guarantees the best performance.
4. CM07 (Competence) Analyse the computational complexity of the algorithmic solutions to develop and implement the one that guarantees the best performance.
5. CM08 (Competence) Ensure the correct functioning of an algorithmic solution in accordance with the requirements of the problem to be solved.
6. KM06 (Knowledge) Recognise the basic concepts of computer logic, structure and programming.
7. KM07 (Knowledge) Describe the basic functioning of computer systems.
8. KM08 (Knowledge) Recognise the methods, systems and technologies specific to computation.
9. SM07 (Skill) Use operating systems and software commonly used in various fields.
10. SM07 (Skill) Use operating systems and software commonly used in various fields.
11. SM08 (Skill) Use algorithmic and data representation structures suitable for problem-solving.

Content

Topic 1: Introduction to computer science
History. Functional structure of the computer. Programs / instructions. Conceptual levels of the computer.

Topic 2: Problem solving: introduction to algorithmics and programming.
Introduction to problem solving. Algorithm concept Phases in the development of algorithms. Programming as an engineering discipline. Software life cycle. Basic elements of an algorithm. Tools for the representation of algorithms. Programming languages. Classification. Language translators: Compilers and interpreters.

Topic 3: Basic concepts and control structures
Definition of variables and constants. Type of fundamental data. Sequential structure. Selection or conditional structures. Iterative or repetitive structures.

Topic 4: Data structures
Unidimensional arrays: strings, tuples and lists.

Topic 5: Subprograms
The concept of a subprogram as an abstraction of operations. Location, neously, scope and visibility. Definition of functions and procedures. Calls to functions and procedures. Modular design descending.

Topic 6: Files
Basic definitions. Input / output of data in files. Types of access to files.

Topic 7: Error control
Types of errors Exceptions and asserts. Preventive programming. Debugging programs.

Topic 8: Introduction to object-oriented programming
Classes and objects. Attributes and methods. Encapsulation. Definition of classes.

Topic 9: Complex data types
Lists: iterators generators, functional paradigm and list comprehensions. Sets. Dictionaries

Activities and Methodology

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Practical classes	13	0.52	CM06, CM07, CM08, KM06, KM07, KM08, SM07, SM08, CM06
Problem classes	26	1.04	CM06, CM07, CM08, KM06, KM07, KM08, SM07, SM08, CM06
Theory Classes	10	0.4	CM06, CM07, CM08, KM06, KM07, KM08, SM07, SM08, CM06
Type: Supervised			
Proyecto de programación	30	1.2	CM06, CM07, CM08, KM06, KM07, KM08, SM07, SM08, CM06
Type: Autonomous			
Preparation classes and personal study	20	0.8	CM06, CM07, CM08, KM06, KM07, KM08, SM07, SM08, CM06
Resolution of self-assessed problems (individual)	46	1.84	CM06, CM07, CM08, KM06, KM07, KM08, SM07, SM08, CM06

The teaching management of the subject will be done through the documentary manager Caronte (http://caronte.uab.cat/), which will be used to view the materials, manage the practices groups, make the corresponding deliveries, see the notes, communicate with the teachers, etc. To be able to use it you have to do the following steps:

1. Register as a user giving the name, NIU, and a photo ID in JPG format. If you have already registered for another subject, you do not need to do it again, you can go to the next step.
2. Register for the type of teaching "Introduction to Programming", giving as a subject code "IniProg" (without quotes).

The teaching of the subject is developed in two sessions of two hours each. In these sessions, five types of teaching activities can be differentiated:

MD1 Exposition of contents in class - Master class: The typical structure of a master class of this type will be the following: in the first place, one will be done Introduction where the objectives of the exhibition and the contents to be discussed will be briefly presented. In order to provide the appropriate context, the presentation will refer to the material exhibited in previous classes, so that the position of these contents is clarified within the general framework of the subject.

MD2 Participatory classes: Joint resolution of the set of problems proposed to students. All the subjects will be accompanied by a relation of problems that the student must try to solve. In this sense, and as the student

progresses in the depth of their knowledge, these problems will be gradually more complex, allowing in this way to clearly appreciate the advantages of using the methodological tools taught during the course. All the problems developed in class and others that may be proposed can be found on the Caronte platform, and will be self-assessed. These activities should allow students to deepentheir understanding and personalize knowledge. The fact that they are self-assessed allows you to adjust the pace of consolidation and reflect on your own learning.

MD3 Tutorials: Hours of free disposal for the student for inquiries about aspects that need additional help from thefaculty.

MD4 Carry out short projects: Completion of practices, larger problems, to deepen in applied aspects of the theory. The practical part of each subject will be completed with at least one practical session, where students will have to solve specific problems of a certain complexity. These projects will be solved in small groups. Each member of the group will have to do a part and put it in common with the rest to have the final solution.

MD5 Assessment activities: See the teaching guide's assessment section

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

Continous Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
1. Partial Theoretical Exam (Tp)	15%	2	0.08	CM06, CM07, CM08, KM06, KM07, KM08, SM08
2. Final Theoretical Exam (Tf)	35%	2	0.08	CM06, CM07, CM08, KM06, KM07, KM08, SM08
3. Delivery of Problems (P)	20%	0	0	CM06, CM07, CM08, KM06, KM07, KM08, SM07, SM08
4. Programming Projects (PPg) - group	20%	0.5	0.02	CM06, CM07, CM08, KM06, KM07, KM08, SM07, SM08
5. Exam Programming Projects (PPi) - individual	10%	0.5	0.02	CM06, CM07, CM08, KM06, KM07, KM08, SM07, SM08

Assessment process and scheduled evaluation activities

Throughout the course the following evaluation activities will be carried out:

Activity	Date	Recovery	Percentage	Minimum note
Theoretical Partial Exam (Tp) Individual	Check Planning	Check Planning	15%	No

Activity				
Final Theoretical Examination (Tf) Individual	Check Planning	Check Planning	35%	Tf >= 5
Delivery of Problems (P) Individual	Each Week	No	20%	No
Programming Projects (PPg) Group	Check Planning	Check Planning	20%	PPg >= 5
Programming Projects Exam (PPi) individual	Check Planning	Check Planning	10%	PPi >= 5

To be able to pass the subject, the result of the weighted sum of the evaluation activities must be greater than or equal to 5, and a minimum grade of 5 must be obtained in activities Tf, PPg and PPi.

Programming of evaluation activities

The dates of evaluation and delivery of works will be published in the documentary manager Caronte and may be subject to changes of programming for reasons of adaptation to possible incidents. The document manager will always be informed about these changes since it is understood that this is the usual platform for exchanging information between teachers and students.

Recovery procedure

The student can submit to the recovery whenever it has been presented to a set of activities that represents a minimum of two thirds of the total grade of the subject.

The theoretical examinations (Tp and Tf) can be recovered in a single examination of recovery to the dates fixed by the coordination of the Degree. The recovery exam will have a percentage of 50% on the final grade. The exam of the examination of recovery, in case of realization, will replace the notes of the partial exams (Tp) and final (Tf) realized during the course.

In accordance with the coordination of the Degree, the Individual Delivery of Problems (P) activity can not be recovered.

Procedure for reviewing the qualifications

Students will have the right to review the theoretical exams (Tp and Tf). The site, date and time of review will be published on the day the notes are published. If the student does not submit to this review, this activity will not be reviewed later. Only in justified cases can a posteriori review of the fixed date and always up to a maximum of 7 calendar days.

There will not be review of the marks of the programming projects (PPg and PPi) because the evaluation is done in front of the students.

"Unique evaluation" system

This subject offers a "unique evaluation" system.

The unique evaluation system will be composed of the following assessment activities:

Activity	Date	Recovery	Percentage	Minimum note

Final Theory Exam (Tf) Individual	Check Planning	Check Planning	50%	Tf >= 5
Programming Projects (PPg) Group	Date of the Final Theory Exam	Date of the recovery exam	30%	PPg >= 5
Programming Projects Exam (PPi) Individual	Date of the Final Theory Exam	Date of the recovery exam	20%	PPi >= 5

To be able to pass the subject, the result of the weighted sum of the evaluation activities must be greater than or equal to 5, and a minimum grade of 5 must be obtained in activities Tf, PPg and PPi.

The recovery system will be the same as for the continuous assessment.

The review of qualifications will follow the same procedure as for the continuous assessment.

<u>Qualifications</u>

Not Evaluable (NA): Any student who submits a practice or a scheduled assessment will have a mark. It will only be deemed not evaluable in the case of not giving any evaluable activity.

Final Mark: It is based on the weighted sum according to the criteria set out in the evaluation activities section. If in any activity the minimum grade is not obtained, the note will come out of the following formula:

Minimum (weighted sum of grades , 4.5)

Grade of honor: Granting a grade of honor registration is the decision of the faculty responsible for the subject. The regulations of the UAB indicate that MH can only be granted to students who have obtained a final grade equal to or greater than 9.00. You can grant up to 5% of MH of the total number of students enrolled. In case the number of students with a grade greater than or equal to 9 is more than 5% of the total numberof students enrolled, students were prioritized according to the following rules (in order):

1. Students whohave achieved a grade equalto or greater than 9makingfewerrecoveries from activities (either because they have suspended the activity or because the grade has gone up).
2. Students with more grades greater than 9 in activities Tp, Tf, and PPi.
3. Students with a better overall score.

<u>Irregularities by the student, copy and plagiarism</u>

Withoutprejudice to other disciplinary measures that they deem appropriate, the irregularities committed by the student that could lead to a variation of the grade of an evaluation act will be scored with a zero. Therefore, copying, plagiarism, cheating, letting copy, etc. in any of the evaluation activities will involve suspending with a zero. The evaluation activities qualified in this way and by this procedure will not be recoverable. If it is necessary to pass any of these evaluation activities to pass the subject, this subject will be suspended directly, without the opportunity to recover it in the same course. In this case, the numerical note of the file will be the lower value between 3.0 and the weighted average of the notes.

<u>Evaluation of repeating students</u>

Thestudents must complete the course as a whole, there is no note from previous courses.

Bibliography

J. Guttag. *Introduction to Computation and Programming Using Python: With Application to Understanding Data Second Edition*. MIT Press. ISBN-10: 9780262529624

S. Chazallet *Python 3. Los fundamentos del lenguaje*. Eni, ISBN-10: 2409006140

E. Matthes. *Python Crash Course: A Hands-On, Project-Based Introduction to Programming*. No Starch Press ISBN-10: 1593276036

M. Myers. *A Smarter Way to Learn Python: Learn it faster. Remember it longer*. Createspace Independent Pub ISBN-10: 1974431479

A. Prieto, A. Lloris, J.C. Torres. *Introducción a la Informática*. Mc Graw-Hill ISBN-10: 8448146247

A. Prieto, B. Prieto. *Conceptos de Informática*. Mc Graw-Hill, Schaum ISBN-10: 8448198573

L. Joyanes Aguilar. *Fundamentos de Programación: Algoritmos, Estructuras de Datos y Objetos*. Mc. Graw-Hill. ISBN-10:8448161114

Software

We will use the latest version of Anaconda which includes Python 3.x and Spyder (https://www.anaconda.com/products/individual)

Language list

Name	Group	Language	Semester	Turn
(PLAB) Practical laboratories	1	Catalan	first semester	morning-mixed
(PLAB) Practical laboratories	2	Catalan	first semester	morning-mixed
(TE) Theory	1	Catalan	first semester	morning-mixed