UAB
Universitat Autònoma de Barcelona

Programming 1

Code: 107888
ECTS Credits: 6

2025/2026

Degree	Type	Year
Computer Engineering	FB	1

Contact

Name: Roberto Benavente Vidal

Email: robert.benavente@uab.cat

Teachers

Francesc Xavier Roca Marva

Albert Gil Lopez

Cristobal Pio Garcia

Teaching groups languages

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

Prerequisites

Since this is an introductory subject, it is assumed that no prior knowledge of the subject exists.

However, it is advisable to have user-level knowledge of some type of platform (Windows, Mac or Linux).

For in-person classroom activities, it is essential to have a laptop to complete the coursework.

Objectives and Contextualisation

This is a general and introductory course to programming. It will delve into the methodological aspects of programming and the learning of a high-level language. Therefore, the general objectives proposed are the following:

- Understanding the software lifecycle: problem analysis (understanding what is being asked of us), design (proposing a solution to the problem), implementation (coding the chosen solution in a programming language), testing (systematically carrying out a test to ensure the correctness of the implemented solution).
- Providing students with the ability to design algorithms for problem-solving by progressively and systematically introducing a rigorous and structured programming methodology, based primarily on the top-down algorithm design technique.

1

- Introducing the student to a real programming language. The goal is for the student to understand the difference between the flexibility of the pseudo-algorithmic notation used in the initial topics and the strict syntax of a real programming language, both in its lexical (valid words in the language), syntactical (rules for combining them), and semantic (signals) aspects.
- Training students to develop programs following style rules aimed at achieving quality programs. These style rules include those that facilitate code comprehension, such as the use of comments, code indentation, and appropriate names for data types, etc.

Learning Outcomes

1. CM04 (Competence) Validate the proposed solutions to problems in the field of computer science according to their requirements and the established conditions.
2. KM06 (Knowledge) Identify the basics of structured, modular and object-oriented programming, as well as development tools and environments for creating high-quality software.
3. KM07 (Knowledge) Identify the simple data types and complex data structures of high-level programming languages, as well as their storage in memory.
4. SM09 (Skill) Analyse the programming needs of an IT system from the standpoint of the customer's need.
5. SM10 (Skill) Use tools for program analysis, design, coding and debugging.
6. SM11 (Skill) Develop graphics engine and gameplay software for videogames that behaves reliably and efficiently according to specifications, while integrating ethical, social, legal and environmental aspects.
7. SM12 (Skill) Demonstrate the ability to work cooperatively in the development of IT applications.

Content

Unit 1: Introduction to computer science

History. Functional structure of the computer. Conceptual levels of the computer.

Unit 2: Problem solving: introduction to algorithms and programming.

Introduction to problem solving. Concept of algorithm. Phases in algorithm development. Programming as an engineering discipline. Software life cycle. Basic elements of an algorithm. Tools for representing algorithms. Programming languages. Classification. Language translators: Compilers and interpreters.

Unit 3: Basic concepts and control structures

Simple data types. Definition of variables and constants. Sequential structures. Conditional structures. Iterative or repetitive structures.

Unit 4: Subprograms. Procedures and functions

The concept of a subprogram as an abstraction of operations. Locality, nesting, scope, and visibility. Function definition. Function calls. Function declaration or prototype. Argument types: input, output, input/output. Passing arguments to functions: by value and by reference.

Unit 5: Complex data types

Array type. One-dimensional arrays. Multidimensional arrays. Strings. Structures. Arrays of structures.

Activities and Methodology

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Practical Labs	12	0.48	CM04, SM09, SM10, SM11, SM12, CM04
Problem-solving lectures	12	0.48	CM04, KM07, SM09, SM10, SM11, CM04
Theory lectures	26	1.04	KM06, KM07, SM10, SM11, KM06
Type: Supervised			
Practical works	25	1	CM04, SM09, SM10, SM11, SM12, CM04
Type: Autonomous			
Classes preparation	20	0.8	KM06, KM07, KM06
Personal study	20	0.8	CM04, KM06, KM07, SM09, SM10, SM11, CM04
Problem-solving assignments	25	1	CM04, SM09, SM10, SM11, CM04

The management of the course's teaching will be carried out through the Caronte document management system (http://caronte.uab.cat/), which will be used to view materials, manage practice groups, submit assignments, view grades, communicate with the teaching staff, etc. To use it, follow these steps:

1. Register as a user by providing your name, ID number, and a passport photo in JPG format. If you've already registered for another subject, you don't need to do so again; you can skip to the next step.
2. Enroll in the "Programming 1" course, using the subject code provided on the first day of class.

The course will be developed using a flipped classroom methodology. This means that students will need to develop some knowledge before the in-person sessions, which will primarily involve practical activities aimed at addressing as many cases as possible to understand the complexity involved in solving a programming problem.

Before class. The basic concepts are covered before the in-person sessions using materials provided a week in advance on the Caronte platform. The materials covered will include written documentation, multimedia content, and learning activities.

Each student should prepare for classes before attending, reviewing the materials as many times as necessary to assimilate the concepts. As a guide, it is recommended to dedicate a number of hours per week equivalent to the class time, approximately 2 to 3 hours per week. For each topic, a questionnaire will be provided to assess the degree of achievement of the content covered.

In class. The objective of the in-person sessions is to move from information to knowledge. We will emphasize creating knowledge applicable to the analysis and solution of specific problems. The class structure will be:

- Review of the basic concepts of the topic to be covered (15 min). The objective is to resolve any doubts that may exist regarding theoretical aspects that are not understood and to monitor the acquisition of basic concepts.
- Problem-solving (remainder of class time). The goal is to experiment with a practical task that allows you to apply the concepts learned to solve challenges and thus share and create knowledge.

After Class. The goal is to consolidate knowledge. The Caronte platform offers a set of assessable, self-correcting problems that should allow students to deepen their understanding of the topics and personalize their knowledge. The self-assessment allows students to adjust the pace of consolidation to each student and foster reflection on their own learning.

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
Final exam	45%	2	0.08	CM04, KM06, KM07, SM09, SM11
Graded in-class activities	10%	0	0	CM04, KM06, KM07, SM09, SM11
Intra-semester exam	15%	2	0.08	CM04, KM06, KM07, SM09, SM11
Practical works	15%	0	0	CM04, SM09, SM10, SM11, SM12
Problem-solving assignments	10%	0	0	CM04, SM09, SM10, SM11
Self-assessment quizzes	5%	6	0.24	KM06, KM07

<u>Scheduled evaluation activities</u>

The subject consists of the following assessment activities:

Activity	Type	Weight	Minimum grade	Recoverable
Intrasemester exam	individual	15%	No	Yes
Final exam	individual	45%	5	Yes
Lab works	group	15%	5	Yes
Self-assessment quizzes	individual	5%	No	No
Problem-solving assignments	individual	10%	No	Yes
Graded class activities	individual	10%	No	No

To pass the course, you must obtain a minimum grade of 5 on the final exam and the Lab works.

<u>Scheduling evaluation activities</u>

The schedule of assessment activities will be announced on the first day of the course and will be made public through Caronte and on the School of Engineering website, in the exams section.

<u>Recovery process</u>

Students may take the make-up exam provided they have completed a set of activities that represents at least two-thirds of the total grade for the subject. In particular, if the student has not taken the final exam, they will not be able to take the make-up exam.

The final exam may be retaken on the date set by the degree coordinator. If a retake exam is taken, the grade for this exam may also replace the grade for the mid-semester exam if the retake exam grade is higher than the mid-semester exam grade.

The assignments can be made up in a second submission on the date specified in the assessment activities calendar, which will be given on the first day of the course and made public through Caronte.

Problem submissions can be made at any time during the course by submitting them again and evaluating them with the auto-corrector.

According to the Degree Coordinator and the School of Engineering management, the following activities cannot be recovered:

- Self-assessment quizzes
- Graded class activities

<u>Qualification review procedure</u>

For exams (mid-semester, final, and make-up exams), a review location, date, and time will be indicated so students can review the activity with the faculty. In this context, students may submit complaints about the activity grade, which will be evaluated by the faculty responsible for the subject. If the student does not attend this review, the activity will not be reviewed later.

For all other activities, if any errors are detected in the correction, you may request a review at any time, up until the date of the make-up exam review.

<u>Grades</u>

Non-Evaluable (NA): A student will be considered non-evaluable (NA) if he/she has not attended any of the following evaluation activities: subject exams (intra-semester, final and recovery), Lab works and evaluable activities in class.

Final grade: It is calculated as the weighted sum of the assessment activities according to the criteria outlined in the assessment activities section.

If a student fails the course because one of the assessment activities does not achieve the minimum required grade, the numerical grade on the transcript will be the lower of 4.5 and the weighted average of the grades. The exceptions are that a grade of "Not Assessable" will be awarded to students who do not participate in any of the assessment activities, and the numerical grade on the transcript will be the lower between 3.0 and the weighted average of the grades if the student has committed irregularities in an assessment activity.

Honors Grade: Awarding an honor roll is the sole discretion of the faculty responsible for the course. UAB regulations state that honor rolls may only be awarded to students who have obtained a final grade equal to or higher than 9.00. Honor rolls may be awarded to up to 5% of the total number of students enrolled.

If the number of students with a grade of 9 or higher exceeds 5% of the total enrollment, the following prioritization criteria will be applied in the order listed below:

1. Students who have completed fewer recovery activities.
2. Students with the highest final grade.
3. Students with the highest grade on the final exam
4. Students with the highest grade on the intra-semester exam

Irregularities committed by students

Without prejudice to other disciplinary measures deemed appropriate, and in accordance with current academic regulations, any irregularities committed by a student that may lead to a grade variation in an assessable activity will be graded with a zero (0). Assessment activities graded in this manner and by this procedure will not be recoverable. If it is necessary to pass any of these assessment activities to pass the course, the subject will be immediately failed, with no opportunity to recover it in the same academic year. These irregularities include, among others:

- the total or partial copying of a practice, report, or any other assessment activity;
- allow copying;
- present group work not entirely completed by the group members (applied to all members, not just those who have not worked);
- unauthorized use of AI (e.g., Copilot, ChatGPT or equivalent) to solve exercises, practices and/or any other evaluable activity;
- present as one's own materials prepared by a third party, even if they are translations or adaptations, and generally works with non-original and exclusive elementsof the student;
- have communication devices (such as mobile phones, smart watches, camera pens, etc.) accessible during individual theoretical-practical assessment tests (exams);
- talk with classmates during individual theoretical-practical assessment tests (exams);
- copying or attempting to copy from other students during theoretical-practical assessment tests (exams);
- use or attempt to use material related to the subject during theoretical and practical assessment tests (exams), when these have not been explicitly permitted.

In future years of this course, students who have committed irregularities in an assessment process will not have any of their completed assessment activities validated, nor will they be eligible for any specific assessment regulations for repeating students.

Uses of AI

For this subject, the use of Artificial Intelligence (AI) technologies is permitted exclusively for support tasks, such as searching for information, interpreting and explaining programs, answering questions, etc. Under no circumstances is AI permitted for programming assignments submitted as part of the assessment. Any work that includes AI-generated fragments will be considered a breach of academic honesty and will result in the application of the regulations regarding irregularities by the student.

Evaluation of repeating students

If the programming block of the subject "Fundamentals of Computer Science" (FI) was passed in the 2024-2025 academic year, you can opt for an evaluation of the subject that will consist of a summary test, plus the grade corresponding to the FI programming block obtained the previous academic year.

If the student successfully completed the "Programming Methodology" course in the previous or current academic year and passed the review exercisesforthat course, he or she may opt for an assessment consisting of a summary test, without having to complete all the assessment activities. The final grade in this case will be 5.

To be eligible for this differentiated assessment, the interested person must request it from the teacher by email (Robert.Benavente@uab.cat / Xavier.Roca@uab.cat) no later than 15 days after the start of classes.

Unique evaluation

This subject does not include the unique evaluation system.

Bibliography

L. Joyanes Aguilar. Fundamentos de programación: algoritmos, estructuras de datos y objetos. 5ª Edición, McGraw-Hill, 2020. (Online version:
https://www-ingebook-com.eu1.proxy.openathens.net/ib/NPcd/IB_Escritorio_Visualizar?cod_primaria=1000193&l
)

L. Joyanes Aguilar, L. Rodriguez Baena, M. Fernández Azuela. Fundamentos de programación: libro de problemas. 2ª Edición, McGraw-Hill, 2003.

E. Valveny, R. Benavente, A. Lapedriza, M. Ferrer, J. García. Programació en Llenguatge C. Amb 56 problemes resolts i comentats. Servei de publicacions de la UAB, 2009.

L. Joyanes, I. Zahonero: Programación en C: metodología, algoritmos y estructuras de datos. 2ª edición, McGraw-Hill, 2005. (Online version: https://elibro.net/es/lc/uab/titulos/50302)

L. Joyanes, A. Castillo, L. Sánchez, I. Zahonero. Programación en C: libro de problemas. McGraw-Hill, 2002.

B.W. Kernighan, D.M. Ritchie. El lenguaje de programación C. 2ª edición, Prentice Hall, 1986.

B.W. Kernighan, R. Pike. La práctica de la programación. Pearson Educación, 2000.

J. Pujol. Algorismes i Programes. Servei de publicacions de la UAB, 1996.

Software

Microsoft Visual Studio: https://azureforeducation.microsoft.com/devtools

Groups and Languages

Please note that this information is provisional until 30 November 2025. You can check it through this link. To consult the language you will need to enter the CODE of the subject.

Name	Group	Language	Semester	Turn
(PAUL) Classroom practices	411	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	412	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	431	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	432	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	451	Catalan	first semester	afternoon
(PAUL) Classroom practices	452	Catalan	first semester	afternoon
(PLAB) Practical laboratories	411	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	412	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	413	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	414	Catalan/Spanish	first semester	morning-mixed

(PLAB) Practical laboratories	415	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	416	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	417	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	418	Catalan/Spanish	first semester	morning-mixed
(PLAB) Practical laboratories	419	Catalan/Spanish	first semester	afternoon
(PLAB) Practical laboratories	420	Catalan/Spanish	first semester	afternoon
(PLAB) Practical laboratories	421	Catalan/Spanish	first semester	afternoon
(PLAB) Practical laboratories	422	Catalan/Spanish	first semester	afternoon
(TE) Theory	41	Catalan	first semester	morning-mixed
(TE) Theory	43	Catalan	first semester	morning-mixed
(TE) Theory	45	Catalan	first semester	afternoon