UAB
Universitat Autònoma de Barcelona

Data Structures

Code: 104347
ECTS Credits: 6

2025/2026

Degree	Type	Year
Data Engineering	OB	2

Contact

Name: Gema Sanchez Albaladejo

Email: gemma.sanchez@uab.cat

Teachers

Gema Sanchez Albaladejo

Daniel Soto Alvarez

Teaching groups languages

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

Prerequisites

The subject does not have any official prerequisite. In any case, it is assumed that the student has studied the previous subjects of Fundamentals of programming and Advanced Programming , as well as Graphs, Topology and Discrete Geometry . Therefore, you are familiar with the basic and advanced programming structures, Orientation a objects and the concept of graph with the different methods of travel on them.

Objectives and Contextualisation

This subject is part of the Subject Representation of the Data and it should be seen as the logical continuation of the Advanced Programming subject and the practical continuation of the subject of Mathematics Discreet. The basic objective is to deepen into the basic data structures introduced in Programming foundations along with the notions of object-oriented programming introduced in Advanced Programming and expand them with other more complex data structures as well as efficient algorithms to navigate them. The concept of recursive algorithm with simple and more complex recursive algorithms such as those related to tree and graphical paths will be introduced

In this way, the training objectives that are proposed for the subject are the following:

- Be able to analyze a complex problem, design an optimum solution, implement it, calculate its cost and test it.
- Understanding and knowing how to use complex data structures such as trees, graphs, etc. and use them correctly and efficiently to solve complex algorithmic problems.

- Provide the student with the ability to design algorithms to solve complex problems, seeing complex algorithms for searching and searching complex data structures. In addition to analyzing the temporal and spatial complexity of them in order to choose the solution that best suits the needs of each moment.
- Introduce the concept of recursion and its application to the recursive complex structures, as well as being able to analyze the complexity of recursive algorithms.
- Program in a real programming language and be able to debug your own programs.
- Develop the programs following some style guidelines aimed at achieving quality programs.

Competences

- Design efficient algorithmic solutions to computational problems, implement them in the form of robust software developments which are structured and easy to maintain, and verify their validity.
- Develop critical thinking and reasoning and know how to communicate it effectively in both your own language and in English.
- Search, select and manage information and knowledge responsibly.
- Work cooperatively in complex and uncertain environments and with limited resources in a multidisciplinary context, assuming and respecting the role of the different members of the group.

Learning Outcomes

1. Choose and apply the most suitable data structures and strategies to solve an informatics problem efficiently.
2. Develop critical thinking and reasoning and know how to communicate it effectively in both your own language and in English.
3. Develop programmes that are well documented, using a good programming style and be able to debug, test and correct them.
4. Search, select and manage information and knowledge responsibly.
5. Work cooperatively in complex and uncertain environments and with limited resources in a multidisciplinary context, assuming and respecting the role of the different members of the group.

Content

0. Introduction
Objectives and presentation of the subject. Review of Advanced Programming.

1. Linoleal data structures . Lists, tails, batteries.

Representation and manipulation of dynamic data structures: review of lists, introduction to piles and queues.

2. Nonlinear data structures. Hash

Techniques of "Hashing". Hash lists. Hash functions. As implemented in python. Dictionaries

3. Recursion and Sorting algorithms

Introduction to recursive algorithms. Bubble method, QuickSort , MergeSort . Recursion Complexity calculation

4. Nonlinear data structures. Graphs
Representations and tours. BFS, DFS, Problem solving with graphs.

5. Nonlinear data structures. Trees.

Definition and representation of a tree. Arbor trails is. Binary Heaps.

Activities and Methodology

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Autonomous	93	3.72	4, 3, 2, 1, 5
face-to-face classes	48	1.92	4, 3, 2, 1, 5

The teaching methodology of the subject is based on the principle that " *programming is the only way to learn to program* " and, therefore, it will be focused primarily on the student's practical work. The face-to-face sessions of Class will be organized to introduce the theoretical contents of the subject, from a very practical perspective starting Examples and exercises and programming problems that will have to be solved directly with your computer . The main objective of the subject is that the student knows how to solve a given problem, efficiently, using complex data structures, if necessary. For this reason, learning will focus on accompanying the student in their task of solving problems based on the previously studied theoretical concepts in an autonomous way. The Python programming language will be mainly used .

On the other hand, there will be a programming project that will have to be developed in a way mainly autonomous throughout the course (with monitoring and control by the teacher in specific sessions) and that It will involve practically integrating almost all the concepts and programming tools introduced in the sessionsface-to-face in the resolution of a complex real problem. In addition, a set of exercises will be proposed that They will have to be resolved individually throughout the course (some of which will be solved and discussed in the sessions) face-to-face) that should serve to understand, integrate and apply the concepts developed in the sessions face-to-face. The activities of the course (face-to-face sessions, problems and practices) will be used mainly Python programming language .

At face-to-face level, class sessions will be organized in four hours a week and will be held in a classroom with Computers to facilitate the practical work of the student. It is necessary for the student to bring their own laptop to class if the classrooms do not have one. The face-to-face sessions will introduce the concepts that are detailed in the syllabus of the subject. In In some cases, explanatory videos or other complementary material may be made available to the student the student will have to see before the class session. Class sessions will have a pretty practical focus with Examples and exercises that will be presented to students to facilitate the comprehension and learning of the concepts explained. These exercises will be done and discussed during the session and will be used to introduce the contents of the subject and see her practical application.

The student will have to complete the face-to-face classes with the autonomous personal work in the accomplishment of the exercises that they are proposing and that they must serve to finish understanding the contents of the subject. Keep in mind that The syllabus of the subject has a logical continuity throughout the course, so that in order to successfully follow one It is necessary to have assimilated what was explained in the previous sessions. Some of these exercises should be delivered from Individual form as part of the evaluation of the subject.

In addition, the students will have to do in groups of 2 a programming project that will be developed of form Autonomous throughout the course outside of the sessions. The programming project will allow one to approach Programming problem of a certain complexity that integrates most of the concepts explained during the course. Durant During the course, some face-to-face sessions will be devotedto the control, monitoring and evaluation of the work done by the student in the programming project

The teaching of the subject will be managed through Caronte (http://caronte.uab.cat/) and the virtual campus (https://cv.uab.cat/).

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
First partial	25% Final grade	2	0.08	2, 1
Problems	20% Final grade	0	0	4, 3, 2, 1
Project	30% Final grade	2	0.08	4, 3, 2, 1, 5
Recovery test	50% Final grade	3	0.12	2, 1
Second partial	25% Final grade	2	0.08	2, 1

The assessment of the subject will take into account three types of evaluation activities: problem delivery, individual assessment and programming project. The final note of the subject is obtained by combining the evaluation of these 3 activities in the following way:

Final Note = (0.2 * Evaluation Problems) + (0.3 * Project) + (0.5 * Individual Evaluation)

- Delivery of problems : this section includes the delivery of the exercises that are proposed during the course and other activities that are carried out in the sessions of problems.

You will have to get one minimum grade of 5 In this activity to be able to pass the subject.
The exercises that are delivered out of time or that have an evaluation of suspended they will be able to recover and return it at any time during the course before the date of the final exam of the subject, with one reduction on the 20% note . The problems will be weighted according to the weight of the topic to the course as a whole, and the number of problems that must be delivered for each subject.

- Individual assessment : this section includes the result of the individual tests that will be done throughout the course. There will be two partial tests that will be done during the class period of the course in class hours and a final test during the official exam period. This final test will be for recovery and only students who have not passed one of the two partials will have to do it. If one of the two partials has been passed, but the other does not, in this test only the part of the subject corresponding to the partial that has not been passed has to be recovered.

You will have to get one minimum grade of 4 in each of the two partial and minimum grade average of 5 to pass the subject.

The final grade will be the average of the two partials: Individual Assessment = (0.5 * Partial1) + (0.5 * Partial2)

- Project : It includes all the work of the programming project. It includes the evaluation of the two deliveries of the project (a partial delivery at half course and the final delivery) and the evaluation of the follow-up of the project that will be done in the face-to-face sessions that correspond. The final grade will be calculated as follows:

Project = (0.2 * Evaluation follow-up project) + (0.3 * Partial Delivery 1) + (0.5 * Final Delivery)

- You will have to get one minimum grade of 4 in the evaluation of Tracking of the project and one minimum note of 5 in the final delivery of the project to be able to approve the project.
- You will have to get one minimum grade of 5 in the project to be able to pass the subject.

- The note of the final delivery of the project it can be recovered if the project's note is> = 3 and the individual assessment note is> = 5.

Not evaluable : A student will be considered non-evaluable (NA) if he does not make at least 50% of the exercises and does not do any of the following evaluation tests: partial 1, partial 2, final recovery test, delivery end of the practice.

Suspended: If the calculation of the final mark is equal to or greater than 5 but the minimum required in one of the assessment activities can not be reached, the final grade It will be suspended and a 4.5 will be placed in the student's record.

Convalidations: Projects from previous years will not be convalidated.

MH: They will give as many tuition fees as they can within the university regulations, starting with the highest notes and as long as the minimum grade is 9.

Revisions: For each assessment activity, a place, date and time of revision will be indicated in which the student will be able to review the activity with the teacher. In this context, claims can be made about the activity note, which will be evaluated by the teachers responsible for the subject. If the student does not submit to this review, this activity will not be reviewed later.

Important note about copies and plagiarisms:
Notwithstanding other disciplinary measures that are deemed opportune, and in accordance with the current academic regulations, the irregularities committed by a student who can lead to a variation of the qualification will be qualified with zero (0). Les Assessment activities qualified in this way and by this procedure will not be recoverable. If you need to pass any of these assessment activities for Approval of the subject, this subject will be suspended directly, without opportunity of recover it in the same course. These irregularities include, among others:

- the total or partial copy of a practice, report, or any other evaluation activity;
- let copy;
- present a group work not done entirely by the members of the group;
- Present as own materials prepared by a third party, though are translations or adaptations, and generally works with non-original and exclusive elements of the student;
- Have communication devices (such as smartphones, smart phones watches , etc.) accessible during the theoretical-practice tests (individual exams).

In these cases, the numerical note of the file will be the lowest value between 3.0 and the weighted average of the notes (and therefore not approved by compensation).
In the evaluation of the delivery of problems and practices, copy detection tools of the program code will be used.

Note on the planning of the evaluation activities:
The dates of continuous assessment and work delivery will be published at the beginning of the course and may be subject to changes of programming for reasons of adaptation to possible incidents. Charon will always be informed about these changes since it is understood that this is the usual platform for exchanging information between teachers and students.

Single evaluation:

This subject does not foresee the single evaluation system.

Use of AI:

Restricted Use: For this course, the use of Artificial Intelligence (AI) technologies is permitted exclusively for support tasks, such as bibliographic or information searches, proofreading texts or translations. Students must clearly identify which parts have been generated with this technology, specify the tools used, and include a critical reflection on how they influenced theprocess and the final result of the activity. Lack of transparency in

the use of AI in this assessable activity will be considered a breach of academic dishonesty and may result in a partial or total penalty in the activity grade, or greater penalties in serious cases.

Bibliography

Problem solving with algorithms and data Structures using Python. Bradley N. Miller and David L. Ranum. Franklin, Beedle and associates, 2005.

Python Programming: an introduction to computer science. John Zelle. Franklin,Beedle and associates, 2004.

Data Structures and Algorithms in python. Michael T. Googrich, Roberto Tamassia, Michael H. Goldwasser. Ed. Wiley. 2013.

https://www.geeksforgeeks.org/python-programming-language/?ref=ghm

Software

Spyder (Anaconda)

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	81	Catalan	first semester	morning-mixed
(PAUL) Classroom practices	82	Catalan	first semester	morning-mixed