Universidad Zaragoza

Academic Year/course: 2024/25

30213 - Data Structures and Algorithms

Syllabus Information

Academic year: 2024/25
Subject: 30213 - Data Structures and Algorithms
Faculty / School: 110 - Escuela de Ingeniería y Arquitectura
326 - Escuela Universitaria Politécnica de Teruel
Degree: 439 - Bachelor's Degree in Informatics Engineering
443 - Bachelor's Degree in Informatics Engineering
ECTS: 6.0
Year: 2
Semester: First semester
Subject type: Compulsory
Module:

1. General information

In this course, the student will improve the ability to design and develop computer programs with emphasis on the identification, design, and definition of Abstract Data Types (ADTs) regardless of their implementation. The student will learn to design and implement ADTs so that they are reusable, efficient and robust, and to implement them guaranteeing these properties. Some of the most frequently used and fundamental ADTs will be presented, for which different implementation alternatives will be studied and compared.

To take this course it is necessary to have prior training in programming, at the level corresponding to that required to pass the course "Programming II". It is also very convenient to have a mathematical training at the level of the course "Discrete Mathematics".

2. Learning results

R6. Define abstract data types (ADTs) regardless of their implementation.

R7. Implement ADTs in a modular programming language.

R8. Implement and use some fundamental ADTs, such as stacks, queues, lists, trees, Hash tables, and graphs.

R9. Compare different ADT implementation alternatives according to their algorithm execution time and memory usage.

R10. Address the modular design of medium-sized programs by identifying, defining, and implementing the necessary ADTs.

R11. Apply basic algorithmic design techniques to problem solving (such as Divide-and-Conquer, backtracking, greedy algorithms).

3. Syllabus

1. Programming with Abstract Data Types (ADTs).
2. Linear ADTs
3. Tree ADTs.
4. Functional data types.
5. Introduction to graphs.
6. Introduction to algorithm design techniques.

4. Academic activities

Lectures: the syllabus will be explained, 30 hours (all hours indicated are approximate)

Problem-solving classes: sessions to solve exercises, design and implementation cases, 15 hours

Personal study of the course, class and practical activities preparation: continuous study and work from the first day of class, 60 hours

Laboratory assignments: 15 hours in the laboratory, in addition to the teamwork required to complete them

Teamwork: resolution of the tasks proposed in the laboratory assignments and delivery of the source code of the resulting programs appropriately designed, implemented, and documented (25 hours each student).

Evaluation tests: 5 hours

5. Assessment system

The course will be evaluated in the global assessment modality, divided into two parts in each evaluation call:

- P1, written exam, a minimum grade of 5.0 will be required to pass and mediate with the rest of the parts. Programming and design problems must be resolved, and conceptual questions answered if requested.
It will be graded with a score from 0 to 10, for which the following will be assessed: the quality and clarity of the answers and proposed solutions, their adaptation to the specifications and restrictions, the quality of the design, the adequate application of the resolution methods, and the time spent solving the exercises.

- P2, practical part, a minimum grade of 5.0 will be required to pass and mediate with the rest of the parts. There will be two alternatives to obtain the grade for this part, here called P2A and P2B.

- P2A: teamwork in laboratory programming assignments.

  Teams will be formed, according to the size and configuration limits indicated by the professors, which must deliver their results for all the programming problems indicated in the course laboratory assignments, complying with the deadlines and instructions given for each of them.

  In case of plagiarism or other irregular practices detected in the laboratory assignments, all students involved will have their P2A evaluation suspended, and will be required to pass the practical programming exam (P2B).

  For each student, the final grade obtained with the P2A programming teamwork will be used as the student's grade for the P2 practical part, unless the student takes an individual exam P2B as described below.

- P2B: Individual laboratory practical programming exam, in each evaluation call.

  In the practical programming exam, the student will be presented with exercises of a similar nature to those carried out in the laboratory programming assignments or those seen in class.

  For those students who take the individual exam P2B, their exam grade will be used as their grade for the practical part of the subject, P2.

In general, each of the programming assignments as well as the individual practical exam will be graded with a grade from 0 to 10, for which the following will be evaluated: the correct operation and performance of the programs according to specifications, the quality of their design, the proper application of the resolution methods, authorship and originality, and the time spent, as well as the ability of the team members to explain and justify the design.

To pass the course it will be necessary to obtain the minimum required grades, both in P1 and P2, and in such case, to calculate the final grade for the course, P1 will weight 70%, and P2 30%.

If the final grades obtained by the student in parts P1 and P2 are both greater than or equal to the minimum grades required in the respective parts, then the student's final grade will be obtained as the weighted sum of their grades in P1 and P2.
Otherwise, the student's grade will be equal to the minimum of 4.0 or the weighted sum of their P1 and P2 grades.

6. Sustainable Development Goals

8 - Decent Work and Economic Growth
16 - Peace, Justice and Strong Institutions