VNIVERSITAT DE VALÈNCIA

Course Guide
34670 Algorithms and data structures

COURSE DATA

Data Subject	
Code	34670
Name	Algorithms and data structures
Cycle	Grade
ECTS Credits	6.0
Academic year	2023 - 2024

Study (s)

Degree	Center	Acad. year	Period
1400 - Degree in Computer Engineering	School of Engineering	2	First term

Subject-matter

Degree	Subject-matter	Character
1400 - Degree in Computer Engineering	11 - Computing and programming	Obligatory

Coordination

Name	Department
BARBER MIRALLES, FERNANDO	240 - Computer Science

SUMMARY

The course "Data Structures and Algorithms" is a subject of the second year of the Degree of Computer Engineering, which covers part of the matter “Programming and Computation”.

This course explores the knowledge and skills in programming seen on the first course in the subjects "Informatics" and “Programming”, giving a more funded and abstract vision of programming. The student capacity in the analysis of the cost of algorithms and in the development of complex algorithms is improved. Also additional abstract data types are studied.

PREVIOUS KNOWLEDGE

Relationship to other subjects of the same degree

There are no specified enrollment restrictions with other subjects of the curriculum.

Other requirements

It is highly desirable that students have taken the courses "Informatics", Programming and Discrete Mathematics.

OUTCOMES

1400 - Degree in Computer Engineering

- G3 - Ability to design, develop, evaluate and ensure the accessibility, ergonomics, usability and security of computer systems, services and applications, and of the information that these manage.
- G4 - Ability to define, evaluate and select hardware and software platforms for the development and implementation of computer systems, services and applications, in accordance with both the knowledge and the specific skills acquired in the degree.
- G8 - Knowledge of basic subject areas and technologies that serve as a basis for learning and developing new methods and technologies, and of those which provide versatility to adapt to new situations.
- G9 - Ability to solve problems with initiative, decision making, autonomy and creativity. Ability to communicate and transmit the knowledge, skills and abilities of a computer engineer.
- R1 - Ability to design, develop, select and evaluate computer applications and systems while ensuring their reliability, safety and quality, according to ethical principles and current legislation and regulations.
- R6 - Knowledge and application of basic algorithmic procedures of computer technology to design solutions to problems, by analysing the suitability and complexity of the algorithms proposed.
- R7 - Knowledge, design and efficient use of the types and structures of data most suitable for solving a problem.
- TI2 - Ability to select, design, implement, integrate, evaluate, build, manage, exploit and maintain hardware, software and network technologies, within adequate cost and quality thresholds.
- TI6 - Ability to design systems, applications and services based on network technologies, including the Internet, the web, e-commerce, multimedia, interactive services and mobile computing.
- C2 - Ability to acquire, obtain, formalise and represent human knowledge in a computable form for solving problems through a computer system in any field, particularly in those related to aspects of computing, perception and action in intelligent environments.

LEARNING OUTCOMES

This course allows to obtain the following results of learning:

1. To handle preconditions, postconditions and to reason on the correction of the programs
2. To identify the temporal and spatial complexity of simple programs
3. To analyze recursive programs
4. To understand advantages and limitations of different alternative data structures and to be capable of selecting the best option in a particular case

DESCRIPTION OF CONTENTS

1. Algorithm specification

1.1 Introduction.
1.2 States, asserts.
1.3 Pre/Post specification (Hoare triplet).
1.4 ADT specification (Abstract Data Type

2. Algorithm efficiency

2.1 Complexity mesure
2.2 Cases analysis.
2.3 Asimptotic notation.

3. Recursive algorithm design

3.1 Mathematical Induction Principles.
3.2 Recursive design.
3.3 Temporal complexity. Recurrence resolution. Characteristic equation.
3.4 "Divide and conquer" paradigm. Quick sort algorithms review.

4. Advanced ADT I: Trees

4.1 Foundations.
4.2 Binary trees. Representation.
4.3. Search binary trees.
4.4. Heaps.

5. Advanced ADT II: Tables

5.1 Foundations.
5.2 Representation.

6. Advanced ADT III: Graphs

6.1 Foundations.
6.2 Representation.
6.3 Graph traversal.

7. Greedy algorithms

7.1 General outline.
7.2 Minimum spanning tree. Prim algorithm.
7.3 Minimum path problem. Dijkstra algorithm.

8. Backtracking algorithms

8.1 General outline.
8.2 Total exploration of the tree.
8.3 Tree prunning.

WORKLOAD

ACTIVITY	Hours	% To be attended
Theory classes	30,00	100
Laboratory practices	20,00	100
Classroom practices	10,00	100
Development of group work	20,00	0
Development of individual work	6,00	0
Preparation of evaluation activities	15,00	0
Preparing lectures	21,00	0
Preparation of practical classes and problem	28,00	0
TOTAL	**150,00**	

TEACHING METHODOLOGY

The course themes will be developed at the classroom based activities providing general as well as integrated view, analysing in details and of major complexity the key aspects and encouraging students´ participation at every point. These activities are complemented by practical activities in order to apply basic concepts and expand them with the knowledge and experience which is gained during the performance of the proposed works. They include the following classroom activities:

- Lessons of problems and questions in classroom
- Sessions of discussions, solving of problems and exercises previously worked by students
- Laboratory practice
- Classroom evaluation through individual questionnaires at the presence of the professorship

Besides the classroom activities the students are to work at individual tasks (out of the classroom) like: monograph works, guided bibliographic search, issues and problems, as well as preparation for lessons and exams. These tasks mainly will be carried out individually in order to upgrade skills of independent work, but in addition will be included some projects which will require the small group (2-4 students) participation with the purpose to build up the capacity for integration into work groups.

The e-learning platform (Aula Virtual) is used in the University of Valencia as a form of communication with students. Students have access through it to the teaching materials used at lessons as well as to problems and exercises to be resolved.

EVALUATION

The evaluation of the course is carried out according to the following scheme:

- Continuous evaluation (N_Continua) based on the degree of participation and involvement in the teaching-learning process, taking into account regular assistance at the planned classroom activities, resolution of issues and problems and of works to be delivered.
- Individual objective test (N_Examenes) consisting of several tests or knowledge tests which include both theoretical and practical issues as well as problems.
- Evaluation of practical activities (N_Practicas) taking into account the achievement of objectives in the lab sessions, resolving problems and projects elaboration.

The final course grade will be calculated according to the following formula:

Final Grade = 20% N_Continua + 50% N_Examenes + 30% N_Practicas

It is required to obtain a minimum grade of 4.5 out of 10 in N_Examenes and N_Practicas to pass the course.

The grade of N_Continua is not recoverable. The grade is maintained in 2º call.

REFERENCES

Basic

- F. Ferri, J. Albert, G. Martín, Introducció a lanàlisi i disseny dalgorismes, Universitat de Valencia, 1999.
- R. Peña, Diseño de programas. Formalismo y abstracción, Prentice-Hall, 3ª Ed., 2005.
- L.R. Nyhoff, TADs Estructuras de datos y resolución de problemas con C++, Prentice Hall, 2ª Ed., 2005.
- H.M. Deitel, P.J. Deitel, C++ Cómo programar, Pearson Educación, 6ª edición, 2009.

Additional

- M.A. Weiss, Data Structures and Algorithm Analysis in C++, 4ª Ed., Pearson (Addison-Wesley), 2014
- G. Brassard, P. Bratley. Fundamentos de algoritmia, Prentice Hall, 1997.
- R.L. Kruse, A.J. Ryba, Data structures and program design in C++, Prentice Hall, 1999
- L. Joyanes, Programación en C++ : Algoritmos, estructuras de datos y objetos MacGraw-Hill, 2ª Ed., 2006.