



Universitat de Lleida

DEGREE CURRICULUM
DATA STRUCTURES

Coordination: GARRIDO NAVARRO, JUAN ENRIQUE

Academic year 2016-17

Subject's general information

Subject name	DATA STRUCTURES			
Code	102010			
Semester	1st Q(SEMESTER) CONTINUED EVALUATION			
Typology	Degree	Course	Typology	Modality
	Double bachelor's degree: Degree in Computer Engineering and Degree in Business Administration and Management	2	COMPULSORY	Attendance-based
	Bachelor's Degree in Computer Engineering	2	COMPULSORY	Attendance-based
ECTS credits	6			
Groups	1GG,3GM			
Theoretical credits	3			
Practical credits	3			
Coordination	GARRIDO NAVARRO, JUAN ENRIQUE			
Department	INFORMATICA I ENGINYERIA INDUSTRIAL			
Teaching load distribution between lectures and independent student work	Work in class (40%) Work out of class (60%)			
Important information on data processing	Consult this link for more information.			
Language	Catalan and Spanish			
Distribution of credits	Juan Enrique Garrido (GG, GM2): 6 ECTS Xavier Domingo (GM1): 3 ECTS Juan Manuel Gimeno (GM3): 3 ECTS			
Office and hour of attention	Appointments made by email			

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
DOMINGO ALBIN, JAVIER JUAN	xdomingo@diei.udl.cat	3	By appointment
GARRIDO NAVARRO, JUAN ENRIQUE	juanenrique.garrido@diei.udl.cat	6	By appointment
GIMENO ILLA, JUAN MANUEL	jmgimeno@diei.udl.cat	3	By appointment

Subject's extra information

Data Structures is a second year course (first semester) within the degree of Software Engineering at Universitat de Lleida. To follow this subject properly, some previous knowledge/skills on programming, Java, and object-oriented programming are recommended.

Data Structures is designed to follow up on Programming II by delving into Object Oriented Programming. Java will be the OOP language used throughout the course.

Data Structures aims to deepen and widen Algorithms and Complexity by discussing algorithms related to data structures in terms of the "Big-Oh" notation.

Data Structures is designed to keep the student's workload as constant as possible throughout the course.

Learning objectives

- To be conversant with the main types of data structures: sequential access, trees, and tables
- To delve into object oriented programming; design and develop interfaces, abstract classes and generics in data structures by using the Java Collections Framework
- To analyse operations and algorithms by using Big Oh notation, and develop more efficient algorithms
- To delve into recursion; design and develop recursive methods to traverse tree and turn these methods into iterative ones
- To design and develop classes which make use of several data structures and aspects related to object oriented programming in order to solve problems

Competences

Cross-disciplinary Competences

- **EPS1.** Capacity to solve problems and prepare and defence arguments inside the area of studies.
- **EPS5.** Capacity of abstraction and of critical, logical and mathematical thinking.

Specific Competences

- **GII-FB3.** Capacity to understand and master the basic concepts of discreet mathematics, logical, algorithmic and computational complexity, and its application to solve engineering problems.
- **GII-CRI6.** Knowledge and application of the basic algorithmic procedures of the computer technologies to design problem solving, analysing the suitability and complexity of the algorithms proposed.

- **GII-CRI7.** Knowledge, design and efficient use of the types and data structure more suitable for solving a problem.
- **GII-CRI8.** Capacity to analyse, design, build and keep safety and efficiency in applications, choosing the paradigm and the most suitable programming languages.

Subject contents

1. Introduction to the Analysis of Algorithms
2. Object Oriented Programming Concepts
 - 2.1 Types, Interfaces and Inheritance
 - 2.2 Introduction to the JCF
3. Sequential Data Structures
 - 3.1 Stacks
 - 3.2 Lists: ArrayList and LinkedList
 - 3.2 Circular and priority queues
4. Arborescent Data Structures
 - 4.1 Definitions and notations
 - 4.2 Review of recursion
 - 4.3 Binary Trees
 - 4.4 Traversals
 - 4.5 Binary Search Trees
 - 4.6 M-ary Search Trees
5. Introduction to Other Data Structures

Methodology

Big-size Group: Theory (3 crédits)

- Classes supported by handnotes
- Always working on examples.

Mid-size Groups: Laboratory (3 crédits)

- Practice on exercises and projects (groups of two)
- Personalized monitoring
- Use of an Integrated Development Environment (Netbeans)

Autonomous Work:

- Study
- Project completion

Development plan

Week	Big-size Group	Mid-size Group	Autonomous Work
1	1. Analysis Algorítmes	Laboratory 1	Study
2	2. OOP Aspects	Laboratory 1	Study
3	2. OOP Aspects	Laboratory 2	Study
4	2. OOP Aspects	Laboratory 2	Study
5	3. Sequential DS	Mini-project 1	Study and project
6	3. Sequential DS	Mini-project 1	Study and project
7	3. Sequential DS	Mini-project 2	Study and project
8	3. Sequential DS	Mini-project 2	Study and project
9	First Midterm		
10	4. Arborescent DS	Laboratory 3	Study
11	4. Arborescent DS	Laboratory 3	Study
12	4. Arborescent DS	Mini-project 3	Study and project
13	4. Arborescent DS	Mini-project 3	Study and project
14	4. Arborescent DS 5. Other DS	Mini-project 4	Study and project
15	5. Other DS	Mini-project 4	Study and project
16	Second Midterm		Study
17	Second Midterm		Study
18	Tutorials		Study and project
19	Recovery Exams		Study

Evaluation

Acr	Descripció	Weight	Minimum Grade	Mandatory	Recoverable	Ind/Grup
Mid1	First Midterm Themes 1, 2, 3 Laboratories 1 i 2	25%	4,0	Yes	Yes	Ind
Mid2	Second Midterm Themes 4 i 5 Laboratory 3 Projects 1 a 4	25%	4,0	Yes	Yes	Ind
Lab	(lab1 + lab2 + lab3) /3	25%	5,0	Yes	Yes	2
Proj	(proj1 + proj2 + proj3 + proj4) / 4	25%	5,0	Yes	Yes	2

Final grade = $0,25 * \text{Mid1} + 0,25 * \text{Mid2} + 0,25 * \text{Lab} + 0,25 * \text{Proj}$

- The recoveries will have the same weights than the normal evaluation activities.
- Recovery activities (midterms, laboratories, projects) won't be to get a better grade if the activity is already passed.
- Passed grades will be maintained for the same course but not between them.
- The evolution of the student will also be taken into account when computing the final grade.
- **To pass the subject all mandatory activities have to be passed.**

Bibliography

William J. Collins. Data Structures and the Java Collections Framework. Third edition. John Wiley & Sons, 2010. USA.

Adam Drozdek. Data Structures and Algorithms in Java. Second edition. Thomson Learning, 2005. USA.

Maurice Naftalin, Philip Wadler. Java Generics and Collections. O'reilly, 2007. USA.

Josep Maria Ribó. Apropament a les estructures de dades del del programari lliure. 2014.

Mark Allen Weiss. Data Structures & Problem Solving Using Java. Fourth Edition. Addison Wesley, 2010. USA.