



Universitat de Lleida

DEGREE CURRICULUM  
**ESTRUCTURES DE DADES**

Coordination: Josep Maria Ribó

Academic year 2013-14

## Subject's general information

<b>Subject name</b>	ESTRUCTURES DE DADES
<b>Code</b>	102010
<b>Semester</b>	1r Q Avaluació Continuada
<b>Typology</b>	Obligatòria
<b>ECTS credits</b>	6
<b>Theoretical credits</b>	3
<b>Practical credits</b>	3
<b>Coordination</b>	Josep Maria Ribó
<b>Department</b>	Informàtica i Enginyeria Industrial
<b>Important information on data processing</b>	Consult <a href="#">this link</a> for more information.
<b>Language</b>	Catalan (Teoria, L2, L3), Castellà (L1)
<b>Office and hour of attention</b>	A concertar per correu electrònic.

Josep M. Ribo, Francisco Jurado

## Subject's extra information

Per tenir èxit a l'assignatura d'Estructures de dades resulta interessant tenir algun dels atributs següents: Sentir un cert entusiasme per la programació i per la resolució de problemes vistos com a reptes intel·lectuals. Tenir gana. No conformar-se amb un aprovat. Voler un excel·lent i treballar CADA DIA per aconseguir-lo (n'hi ha per a tots). Afrontar els problemes i les pràctiques des del primer moment en què es publiquin. No esperar al darrer dia.

L'assignatura d'Estructures de dades s'ubica al tercer quadrimestre del grau en E. Informàtica. És la continuació natural de Programació 2, perquè aprofundeix en la disciplina de la programació i, especialment, en el paradigma de la Programació Orientada a Objectes (POO).

\*Complementa Matemàtica discreta (també del tercer semestre) perquè un dels seus objectius és arribar a implementar un dels objectes que es descriuen teòricament a aquesta assignatura: els grafs

\*Complementa també Algorismes i complexitat perquè descriu algorismes associats a estructures de dades i discuteix la seva eficiència.

\*Dóna eines que seran usades a les assignatures de Bases de dades i Enginyeria de software (en quadrimestres posteriors). En particular, Estructures de dades descriu algunes implementacions d'índexos i de fitxers directes que s'usaran a Bases de dades i també presenta el paradigma de la POO i algun patró de disseny, aspectes en els quals s'aprofundirà a Enginyeria de software.

## Learning objectives

See competences

## Competences

Degree-specific competences

- Knowledge, design and efficient use of the most appropriate data type and structure to resolve a problem.

Goals

- Chooses between list implementations
- Implements a linked list
- Chooses a map implementation
- Implements an open hash map
- Implements a closed hash map
- Designs classes that combines various data structures
- Designs recursive algorithms on trees
- Designs a Graph class

- Knowledge and application of basic computer technologies' algorithmic procedures to design solutions to problems, by analysing the suitability and complexity of the proposed algorithms.

Goals

- Knows the cost of graph algorithms

- Designs recursive algorithms on trees
- Ability to understand and master the basic concepts of discrete mathematics, logic, algorithm and computational complexity, and their application to the resolution of engineering problems.

## Goals

- Designs a graph class
- Applies the big-O notation to state the cost of an algorithm
- Ability to analyse, design, build and maintain applications in a general, secure and efficient way, by choosing the most appropriate programming paradigm and language.

## Goals

- Designs polymorphic type hierarchies
- Is aware of free software
- Designs classes with exceptions

## Subject contents

1

### Subject contents

#### 1. introduction: The object Oriented Paradigm (OOP)

##### 1.1 Exceptions

##### 1.2 Generalizations, inheritance, polymorphism

##### 1.3 Generic types

#### 2. Sequential data structures: lists

##### 2.1 What are containers?

##### 2.2 Containers in Java Collection Framework

##### 2.3 Iterables and iterators. Specification and usage.

##### 2.4 Collections: specification and usage.

##### 2.5 Lists. Specification

##### 2.6 Lists. Implementations.

##### 2.7 LinkedList<T>

#### 3. Tree data structures

##### 3.1 Definitions, notation, properties and traversals.

##### 3.2 Binary trees. Specification.

3.3 Binary trees. Implementation.

4. Direct access data structures: Maps

4.1 What are maps?

4.2 Maps. Specification.

4.3 Maps: implementation strategies.

4.4 Maps: Implementation with hash functions.

4.5 Maps: Implementation with binary search trees.

4.6 Maps: Implementation with B-trees.

## Methodology

- Classes de teoria (2 h. per setmana): Classe magistral amb participació dels estudiants
- Classes de laboratori (2 h. per setmana): En general, es desenvoluparan de la manera següent:
  - Una hora de problemes resolts pel professor.
  - Una hora de laboratori amb problemes que resoldran els estudiants amb la supervisió del professor.
  - Algunes classes de laboratori estan orientades al projecte XPR. En aquelles classes, s'explica un capítol del projecte i es comença a treballar.

## Development plan

- Teoria i problemes:
  - Tema 1: Setmanes 1 a 3
  - Tema 2: Setmanes 4 a 7
  - Tema 3: Setmanes 7 a 11
  - Tema 4: Setmanes 12 a 15
- Projecte XPR:
  - Capítols 1, 2 i 3: Lliurament: setmana 7
  - Capítol 4: Lliurament: setmana 12
  - Capítol 5: Lliurament: Durant festes de Nadal
  - Capítol 6: Lliurament: setmana 18

## Evaluation

L'avaluació es compon de les activitats següents (les dates no són oficials, tot i que mirarem de mantenir-les):

- Examen parcial Novembre: Temes 1, 2, 3.1. 25% nota final.
- Examen parcial Gener: Temes 3 i 4. 25% nota final.
- Projecte: eXPertRank (XPR). 25% nota final.
- Laboratori de problemes. Problemes fets a les hores de laboratori que es lliuraran durant el curs. 20% de la nota final.
- Nota de participació. 10% de la nota final.

### Consideracions addicionals sobre l'avaluació

- La setmana de recuperacions es podran recuperar els exàmens parcials 1 i 2 i també qualsevol part del projecte XPR que no hagi estat lliurada durant el curs o que hagi estat considerada incorrecta pels professors.
- Les parts del projecte XPR que es lliurin fora de termini (particularment, durant la setmana de recuperacions) tindran una penalització en la seva avaluació.

- La nota mínima que cal obtenir als parcials 1 i 2 per tal de poder aprovar l'assignatura sense haver-los de repetir la setmana de recuperacions és 4. El professor podrà establir excepcions a aquesta regla si la progressió positiva d'un estudiant així ho suggereix.

## Bibliography

### Basic bibliography:

- William Collins: Data Structures and the Java Collections Framework ISBN: 978-0-470-48267-4
- Adam Drozdek: Data Structures and Algorithms in Java [Paperback] ISBN: 981-4239-23-2
- Mark Guzdial, Barbara Ericson: Problem Solving with Data Structures Using Java: A Multimedia Approach. Prentice Hall
- Mark A. Weiss: Data Structures and Problem Solving Using Java (3a o 4a edició. Preferiblement, 4a).Addison Wesley. ISBN: 9780321541406

### Complementary bibliography:

- Roberts, Eric S.: The Art & science of Java : an introduction to computer science / Eric S. Roberts Pearson/Addison Wesley, cop. 2008. ISBN: 0321486129 (Per repassar els aspectes fonamentals de Java.
- Kurt Mehlhorn, Peter Sanders: Algorithms and data structures the basic toolbox Springer ISBN: 9783540779780 (Llibre amb aspectes més teòrics)
- Joan Gimbert i altres: Apropament a la teoria de grafs i els seus algorismes. Edicions de la UdL, n. 23 (Pel projecte en què usarem grafs).