DEGREE CURRICULUM

PROGRAMACIÓ I

Coordination: Assignatura que s'imparteix durant el primer semestre del primer curs de la titulació. Correspon a la Matèria "Informàtica" dins del Mòdul de "Formació Bàsica".

Academic year 2013-14

INTRODUCTION TO PROGRAMMING I 2013-14

Subject's general information

Subject name	PROGRAMACIÓ I
Code	102000
Semester	1r semestre Avaluació Continuada
Typology	Troncal
ECTS credits	6
Groups	1 Grup Gran en el GEI, 3 Grups reduïts en el GEI, 1 grup de Docència repetida en el GEI. 1 Grup en el GEIADE
Theoretical credits	2
Practical credits	4
Coordination	Assignatura que s'imparteix durant el primer semestre del primer curs de la titulació. Correspon a la Matèria "Informàtica" dins del Mòdul de "Formació Bàsica".
Department	Informàtica i Enginyeria Industrial
Teaching load distribution between lectures and independent student work	Cada setmana l'estudiant assiteix a 2 hores presencials amb Grup Gran i 2 hores presencials amb Grup Reduït. Les sessions amb Grup Reduït s'imparteixen al laboratori. A les sessions amb Grup Gran presentem els conceptes i les estructures algorísmiques utilitzades en l'assignatura. Per a cada estructura algorísmica es proposa una col.lecció d'exercicis de programació. El treball autònom de l'estudiant consisteix en el disseny i implementació de les solucions a la llista d'exercicis proposats. A les corresponent sessions de Grup Reduït s'analitzen les solucions dissenyades i es resolen els problemes trobats. Finalment, a les sessions de Grup Reduït es dona suport a les pràctiques obligatories que ha de desenvolupar l'estudiant al llarg de l'assignatura de forma autònoma.
Important information on data processing	Consult this link for more information.
Language	Catalan
Distribution of credits	2 crèdits per a la presentació dels conceptes i les estructures algorísmiques utilitzades en l'assignatura. 2 crèdits per a la resolució de problemes (pràctics). 2 crèdits de laboratori (implementació de les solucions).
Office and hour of attention	Josep Argelich Romà A concretar per correu electrònic Maria Teresa Alsinet Bernadó Thursday and Friday from 12:00 to 13:00.

INTRODUCTION TO PROGRAMMING I 2013-14

Josep Argelich Romà
Maria Teresa Alsinet Bernadó

Subject's extra information

Suggestions

Solve the problems and programming exercises that are daily proposed allow to achieve the goals.

The course as part of the academic plan

The main objective of this course is to design algorithms, then implement them in a compilable programming language. Specifically, the imperative language C is chosen and the problems to solve are mainly processing sequences. Under this framework, the course contents are structured in four blocks. The first presents the basic instructions of the algorithmic language used throughout the course as well as the programming language C. The second shows how to design "simple" algorithms by identifying the problem to be solved and the application of direct and sequential access patterns, in addition, we present the non-basic data types that support them. The third shows the descendant design technique which can address more "complex" problems from "simple" problems. Finally, the fourth part of the course focuses on the management of dynamic memory in C and its integration with the algorithmic schemes studied throughout the course.

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

- Design and implement algorithms to solve complex problems in a structured and efficient way.
- Design and implement the operations associated with the structures identified
- Integrate the data design in the application design.
- Design and implement appropriate data structures to represent information of each problem.

- Basic knowledge of computer use and programming, operative systems, data bases and computer programmes with applications in engineering.

Goals

- To use a software development environment based on a high-level programming language.
- Dynamic memory management using the features of its own programming language.
- Design and implement basic algorithmic solutions using the refining design technique.
- Identify the type of problem and apply the appropriate algorithmic strategy.
- Design and implement iterative algorithms.
- Design and implement algorithmic structures suitable for solving different types of problems.

Degree-transversal competences

- Be motivated by quality and continual improvement.

INTRODUCTION TO PROGRAMMING I 2013-14

Goals

- To design and implement appropriate data structures to represent information of each problem.
- Learn to design efficient algorithms and to implement them in a compilable programming language.

Subject contents

Topic 1. Basic algorithmic structures

1.1 Constants, variables, basic types and valid expressions

1.2 Assignment, sequential composition, alternative composition and iterative composition

1.3 Programming Environment.

Topic 2. Iterative design of programs

2.1 Sequential Access

2.1.1 Algorithmic schemes for sequence processing

2.1.2 Algorithmic schemes for search in sequences

2.2 Direct access. Tables

2.2.1 Sequential tables

2.2.2 Direct tables

2.2.3 Classic sorting algorithms

Topic 3. Non-basic data processing

3.1. Descendant design of algorithms

3.2 Procedures and Functions

3.3 Tuples

Topic 4. Memory management in C

4.1 Management of memory addresses

4.2 Allocation and freeing of memory blocks

Topic 1. Basic algorithmic structures

1.1 Constants, variables, basic types and valid expressions

1.2 Assignment, sequential composition, alternative composition and iterative composition

1.3 Programming Environment.

Topic 2. Iterative design of programs

2.1 Sequential Access

2.1.1 Algorithmic schemes for sequence processing

2.1.2 Algorithmic schemes for search in sequences

2.2 Direct access. Tables

2.2.1 Sequential tables

2.2.2 Direct tables

2.2.3 Classic sorting algorithms

Topic 3. Non-basic data processing

3.1. Descendant design of algorithms

3.2 Procedures and Functions

3.3 Tuples

Topic 4. Memory management in C

4.1 Management of memory addresses

4.2 Allocation and freeing of memory blocks

Methodology

Els continguts del curs s'estructuren en quatre grans blocs. El primer presenta les instruccions bàsiques del llenguatge llenguatge de programació ANSI C/C++. El segon mostra com dissenyar algorismes per al tractament de seqüències mitjançant la identificació del problema a resoldre i l'aplicació d'esquemes d'accés seqüencial i directe, a més, es presenten els tipus de dades no elementals que els donen suport. El tercer mostra la tècnica de disseny descendent d'algorismes la qual permet abordar problemes generals a partir de la seva descomposició en problemes més concrets. Finalment, el quart bloc del curs es centra en la gestió de la memòria dinàmica en ANSI C/C++ i la seva integració en els esquemes algorísmics estudiats al llarg del curs.

Per a cada bloc es proposa un col.lecció de problemes que l'estudiant haurà d'abordar de forma autònoma i supervisada a les sessions de laboratori.

Development plan

Descripció
Presentació de l'assignatura. Introducció a la matèria de l'assignatura: Algorismes i programes Tema 1. Estructures algorísmiques bàsiques 1.1 Constants, variables, tipus elementals, i expressions vàlides 1.2 L'assignació, la composició seqüencial, la composició alternativa i la composició iterativa 1.3 Entorn de programació.
Total hores presencials: 10 h (3 sessions GG i 2 sessions GP)
Total hores no presencials: 15 h

Descripció: Descripció i organització de la primera pràctica obligatòria.
Total hores presencials: 2h (1 sessió GP)
Total hores no presencials: 3 h

Descripció
Tema 2. Disseny de programes iteratius 2.1 Accés seqüencial 2.1.1 Esquemes algorísmics de tractament de seqüències 2.1.2Esquemes algorísmics de cerca en seqüències 2.2 Accés directe. Les taules 2.2.1 Tractament seqüencial de taules 2.2.2 Tractament directe de taules. Algorismes d'ordenació clàssics. Preparació de la prova escrita 1. Autoavaluació.
Total hores presencials: 18 h (5 sessions GG i 4 sessions GP)
Total hores no presencials: 27 h

Descripció
Activitats d'avaluació: - Prova escrita 1 (25%) -Lliurament de la primera pràctica obligatòria (15%)

Total hores presencials: 2h
Total hores no presencials: 3 h

Descripció
Tema 3. Tractament de dades no elementals 3.1.Disseny descendent d'algorismes 3.2 Accions i funcions 3.3 Tuples.
Total hores presencials: 18 h (5 sessions GG i 4 sessions GP)
Total hores no presencials: 27 h

Descripció
Descripció i organització de la 2a pràctica obligatòria. Resolució d'aspectes concrets de la pràctica.
Total hores presencials: 2h (1 sessió GP)
Total hores no presencials: 3 h

Descripció
Tema 4. Gestió de memòria en C 4.1 Gestió d'adreces de memòria 4.2 Assignació i alliberació de blocs de memòria.
Total hores presencials: 4 h (1 sessió GG i 1 sessió GP)
Total hores no presencials: 6 h

Descripció
Activitats d'avaluació: - Prova escrita 2 (35%) -Lliurament de la segona pràctica obligatòria (25%)
Total hores presencials: 2h
Total hores no presencials: 3 h

Descripció
Recuperació/millora de la nota obtinguda a les proves escrites 1 i 2.
Total hores presencials: 2h
Total hores no presencials: 3 h

Evaluation

Prova Escrita 1:

Goals:

- Design and implement algorithmic structures suitable for solving different types of problems.
- Design and implement iterative algorithms.
- Identify the type of problem and apply the appropriate algorithmic strategy.
- Design and implement appropriate data structures to represent information of each problem.

Criteria

To pass the course, the mark obtained in the written test must be >=3.

Realisation Individual

Character Compulsory

Task Written test

Percentage 25%

Prova Escrita 2:

INTRODUCTION TO PROGRAMMING I 2013-14

Objectius:

Aprendre a dissenyar i implementar les estructures de dades adecuades per representar la informació pròpia de cada problema.
Dissenyar i implementar estructures algorísmiques adequades per resoldre les diferents tipologies de problemes.Dissenyar i implementar algorismes iteratius.
Identificar la tipologia del problema i aplicar l'estratègia algorísmica adequada. Dissenyar i implementar solucions algorísmiques bàsiques utilitzant la tècnica de disseny descendent.
Utilitzar les funcionalitats pròpies dels llenguatges de programació per la gestió de memòria dinàmica.
Dissenyar i implementar algorismes per resoldre problemes complexes de forma estructurada i eficient.
Dissenyar i implementar les operacions associades amb les estructures identificades. Dissenyar i implementar estructures de dades adequades per representar la informació pròpia de cada problema.

Criteris: La prova escrita s'avaluarà sobre 10 punts. Per aprovar l'assignatura la nota obtinguda en aquesta prova escrita ha de ser >=3. Aquesta prova escrita representarà un 35% de la nota final.

Observacions: Si la nota obtinguda en aquesta prova escrita és >=5, aleshores la nota obtinguda podrà actuar com a recuperació/millora de la primera prova escrita, el pes de la qual és del 25%.

Pràctica Obligatòria 1:

Goals

- Learn to design efficient algorithms and to implement them in a compilable programming language.
- Design and implement algorithmic structures suitable for solving different types of problems.
- Design and implement iterative algorithms.
- Identify the type of problem and apply the appropriate algorithmic strategy.
- To use a software development environment based on a high-level programming language.
- Design and implement algorithms to solve complex problems in a structured and efficient way.
- Design and implement the operations associated with the structures identified
- Integrate the data design in the application design.
- Design and implement appropriate data structures to represent information of each problem.

Criteria

To pass the course, the mark obtained in the written test must be >=3. If the practice fails, the note can be recovered from the recovery period (week number 20).

Observations

Collection of programming exercises. Performing in groups of maximum 2 people. Presentation compulsory.

Pràctica Obligatòria 2:

Goals

- To design and implement appropriate data structures to represent information of each problem.
- Learn to design efficient algorithms and to implement them in a compilable programming language.
- Design and implement algorithmic structures suitable for solving different types of problems.
- Design and implement iterative algorithms.
- Identify the type of problem and apply the appropriate algorithmic strategy.
- Design and implement basic algorithmic solutions using the refining design technique.
- Dynamic memory management using the features of its own programming language.
- To use a software development environment based on a high-level programming language.
- Design and implement algorithms to solve complex problems in a structured and efficient way.
- Design and implement the operations associated with the structures identified
- Integrate the data design in the application design.
- Design and implement appropriate data structures to represent information of each problem.

Criteria

To pass the course, the mark obtained in the written test must be >=3. If the practice fails, the note can be recovered from the recovery period (week number 20).

Observations

Collection of programming exercises. Performing in groups of maximum 2 people. Presentation compulsory.

Recuperació de les proves escrites: Si la nota final obtinguda en l'assignatura és <5, aleshores l'estudiant pot optar a millorar/recuperar el 60% que representen les proves escrites.Una única prova escrita que serà avaluada sobre 10 punts. La nota obtinguda substituirà la nota de les dues proves escrites del curs. La nota de la prova representarà el 60% de la nota final.

Goals

- To design and implement appropriate data structures to represent information of each problem.
- Design and implement algorithmic structures suitable for solving different types of problems.
- Design and implement iterative algorithms.
- Identify the type of problem and apply the appropriate algorithmic strategy.
- Design and implement basic algorithmic solutions using the refining design technique.
- Dynamic memory management using the features of its own programming language.
- Design and implement the operations associated with the structures identified
- Design and implement appropriate data structures to represent information of each problem.

Criteria

The mark obtained in this written test will replace the mark of the two written tests of the course.

Bibliography

Basic References:

- J. Castro, F. Cucker, X. Messeguer, A. Rubio, L. Solano and B.Valles. *Curs de Programació*. McGraw-Hill, 1992.
- J.L. Balcázar. *Programación Metódica*. McGraw-Hill, 1993.
- G. Brassard and P. Bratley. *Fundamentosde Algoritmia*. Prentice Hall, 1997.
- L. Joyanes. Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos. McGraw-Hill, 2003.

ANSI C and C++:

- H.M. Deitel and P.J. Deitel. *ComoProgramar en C/C++*. Prentice-Hall, segunda edición, 2002.
- B.W. Kernighan and D.M. Ritchie. *Ellenguaje de programación C*. Prentice-Hall, segundaedición, 1991.
- B.W. Kernighan and R. Pike. *ThePractice of Programming*. Addison-Wesley, 1999.
- B. Stroustrup. Programming -- Principles and Practice Using C++.Addison Wesley, 2008.
- B. Stroustrup. *El lenguaje deprogramación C++*. Edición especial. Addison Wesley, 2002.
- F.Xhafa; P. Vázquez, J. Marco, X. Molinero and A. Martín. Programación en C++ paraingenieros. Paraninfo, 2006.
- L. Joyanes. Programación en C++. McGraw-Hill, 2006.

Other references:

- E.W.Dijkstra and W.H.J. Feijen. *A method of Programming.*Addison-Wesley, 1988.
- J.Esakov and T. Weiss. *Data Structures. An AdvancedApproach Using C*. Prentice-Hall, 1989.

- A. Kaldewaij. *Programming: The Derivation ofAlgorithms*. Prentice-Hall, 1990.