Course guide
230906 - PRD - Programming and Data Structures

Last modified: 06/06/2024

Unit in charge: Barcelona School of Telecommunications Engineering
Teaching unit: 701 - DAC - Department of Computer Architecture.

Degree: BACHELOR'S DEGREE IN ELECTRONIC ENGINEERING AND TELECOMMUNICATION (Syllabus 2018). (Compulsory subject).

Academic year: 2024 **ECTS Credits:** 6.0 **Languages:** Catalan, Spanish

LECTURER

Coordinating lecturer: JORDI PERELLO MUNTAN

Others: Segon quadrimestre:
SILVIA LLORENTE VIEJO - 13
JORDI PERELLO MUNTAN - 11, 12, 13

PRIOR SKILLS

In order to follow the subject appropriately, it is strongly recommended to have passed the previous programming subject of the degree (Algorithms and Programming), as many of the programming concepts explained there will be assumed already known.

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:
CE1. (ENG) GREELEC: Capacitat per a la resolució dels problemes matemàtics que poguin plantejar-se a l'enginyeria. Aptitut per aplicar els coneixements sobre àlgebra lineal, geometria, geometria diferencial, càlcul diferencial i integral, equacions diferencial i en derivades parcials, mètodes numèrics, algorítmica numèrica, estadística i optimització. (Mòdul de formació bàsica).

Generical:
CG3. (ENG) GREELEC: Coneixmetn de matèries bàsiques i tecnològíes que el capacitin per a l'aprenentatge de nous mètodes i tecnologies, així com que el dotin d'una gran versatilitat per adaptar-se a noves situacions.

Transversal:
CT6. (ENG) GREELEC:APRENENTATGE AUTÒNOM: Detectar deficiències en el propi coneixement i superarles mitjançant la reflexió crítica i l'elecció de la millor actuació per ampliar coneixements.

Basic:
CB5. (ENG) GREELEC: Que els estudiants pugin desenvolupar habillitats d'aprenentatge per empendre estudis superiors amb un alt grau d'autonomia.

TEACHING METHODOLOGY

Expository method / Lecture class
Participative lecture class
Laboratory session
Cooperative work
Autonomous work
Problem/project-based learning

LEARNING OBJECTIVES OF THE SUBJECT

Subject objectives:

1. The student should be able to efficiently implement programs of moderate complexity in C programming language, using the debugger in order to detect and fix errors occurring during the execution of the program when necessary.

2. The student should understand how the system memory is used throughout the execution of a program in C, as well as the utilization of the existing library functions for managing dynamic memory.

3. The student should be able to efficiently implement and manage basic dynamic data structures, both linear (lists, stacks, queues) and non-linear (hash tables) ones.

4. The student should understand bitwise operators in C, being able to use them in basic use cases.

Learning results:

1. The student knows the basic syntax of the C programming language, and is able to use it to efficiently implement the requested programs.

2. The student is aware of the function-based modular programming benefits in C, and is capable to implement previously specified functions using references (i.e., pointers) when necessary.

3. The student is aware of the limitations of static data structures and appreciates the benefits of dynamic data structures, knowing their basic characteristics.

4. The student is able to efficiently implement and manage dynamic data structures, both linear (lists, stacks, queues) and non-linear (hash tables) ones.

5. The student is able to employ bitwise operators for basic use cases.

STUDY LOAD

Type	Hours	Percentage
Self study	85,0	56.67
Hours small group	26,0	17.33
Hours large group	39,0	26.00

Total learning time: 150 h

CONTENTS

1. Introduction to C programming language

Description:
Definition of variables and constants. Data types and type modifiers. Arithmetic, relational, logical and assignment operators in C. Concepts of expression and sentence. Conditional and iterative flow control sentences. Structured datatypes. Examples.

Full-or-part-time: 18h
Theory classes: 9h
Self study : 9h

2. Functions

Description:
Introduction to modular programing in C using functions. Function declaration and definition. Pass by value and pass by reference of parameters to a function. Pointers. Sorting and searching algorithms. Examples.

Full-or-part-time: 19h
Theory classes: 9h
Self study : 10h

3. Dynamic memory management

Description:
Memory types of a program written in C. Library functions for dynamic memory management. Basic use cases: dynamic vectors, array lists.

Full-or-part-time: 10h 30m
Theory classes: 4h 30m
Self study : 6h

4. Dynamic data structures

Description:
Presentation of basic dynamic data structures, both linear (lists, stacks, queues) and non-linear (hash tables). Implementation of linked lists as the basic component enabling the rest of dynamic data structures that will be subsequently presented. Implementation of stacks and queues. Implementation of hash tables. Concept of hash function. Implementation of basic hash functions. Examples.

Full-or-part-time: 25h
Theory classes: 9h
Self study : 16h

5. Bitwise operations

Description:
Number base conversion (decimal to binary, octal and hexadecimal). Bitwise operators. Bit masks. Basic examples.

Full-or-part-time: 12h 30m
Theory classes: 4h 30m
Self study : 8h

Laboratory sessions

Description:
Session 0.- Introduction and installation of the laboratory environment
Session 1.- Initial programs in C programming language
Session 2.- Conditional and iterative flow control sentences
Session 3.- Data structures: structs, vectors and matrices
Session 4.- Pass by value of parameters to functions
Session 5.- Pass by reference of parameters to functions
Session 6.- Dynamic memory management in C
Session 7.- Doubly linked lists
Project assignment 1: First laboratory working session
Project assignment 1: Second laboratory working session
Project assignment 2: Laboratory working session
Laboratory exam

Full-or-part-time: 60h
Laboratory classes: 24h
Self study : 36h

Midterm exam of the course

Description:
Midterm exam of the course.

Full-or-part-time: 2h
Laboratory classes: 2h

Final exam of the course

Description:
Final exam of the course.

Full-or-part-time: 3h
Theory classes: 3h

GRADING SYSTEM

Laboratory: 35% (= 60% project assignments + 40% laboratory exam)
Midterm exam: 15%
Final exam: 50%

Laboratory session attendance is compulsory: unjustified absences can impact negatively on the student's laboratory mark.

It is only reassessable the theoretical part of the subject. Hence, the subject's mark after reassessment will be obtained from the reassessment exam's mark (65%) plus the laboratory mark from the previous evaluation (35%).

EXAMINATION RULES.

It is strictly forbidden to bring lecture notes and programmable devices (mobile phone, laptop, tablet, etc.) during the subject's midterm, final, or reassessment exams.

BIBLIOGRAPHY

Basic:
- Jiménez, M.; Otero, B. Fundamentos de ordenadores: programación en C [on line]. Barcelona: Iniciativa Digital Politècnica, 2013 [Consultation: 14/10/2022]. Available on: http://hdl.handle.net/2099.3/36593. ISBN 9788476539958.
- Joyanes, L.; Zahonero, I. Programación en C: metodología, algoritmos y estructuras de datos. 2a ed. Madrid: McGraw-Hill, 2005. ISBN 8448198441.

Complementary:
- Kernighan, B.W.; Ritchie, D.M. El lenguaje de programación C. 2a ed. México D.F.: Prentice-Hall Hispanoamericana, 1991. ISBN 9789688802052.