DEGREE CURRICULUM

COMPUTER SCIENCE BASICS

Coordination: Josep Lluís Lérida

Academic year 2014-15

COMPUTER SCIENCE BASICS 2014-15

Subject's general information

Subject name	COMPUTER SCIENCE BASICS
Code	102109
Semester	2n Q
Typology	Continuous Assessment
ECTS credits	6
Theoretical credits	50
Practical credits	50
Coordination	Josep Lluís Lérida
Office and hour of attention	Contact by email with the teacher: jlerida@diei.udl.cat valenti.pardo@udl.cat jmsola@diei.udl.cat
Department	Informàtica i Enginyeria Industrial
Teaching load distribution between lectures and independent student work	40% Classroom 60% Autonomous Work
Modality	Presencial
Important information on data processing	Consult this link for more information.
Language	Catalan
Degree	Degree in Automation and Industrial Electronic Engineering Degree in Mechanical Engineering
Distribution of credits	50% Lectures and problems 50% Practices
Office and hour of attention	Contact by email with the teacher: jlerida@diei.udl.cat valenti.pardo@udl.cat jmsola@diei.udl.cat
E-mail addresses	jlerida@diei.udl.cat valenti.pardo@udl.cat jmsola@diei.udl.cat

COMPUTER SCIENCE BASICS 2014-15

Josep Lluis Lerida Monso
Valentí Pardo Casanovas
J. M. Sola Gimeno

Learning objectives

The instrumental nature of the computer programming course make it very useful in many areas and professional fields. The computer programming provides a variety of specific solutions for the professional development in the field of Industrial Engineering, but also plays an essential role in the area of process engineering, control systems and automation. The main objective for this course is that students learn to design and implement on a programmable device, efficient and quality solutions to different types of problems. In addition to manage the programming environment and the solving techniques, this course pretend to introduce the students to the basic components of a programmable device as well as to acquire the conception and practice of how to write a program, compile and run knowledge.

This overall objective can be divided into the following more specific objectives:

1. Identify the basic components of a computer or programmable device, its functionality and the process of interaction with the other components.
2. Learn the utility of E/S alphanumeric codes and how the computer recognizes and uses them.
3. Learn how current programmable devices perform numerical representation and basic operations.
4. Identify the basic characteristics of a high-level language and understand the processes that must be taken in a programmable device to compile or interpret a program and run it.
5. Learn and understand the syntax and semantics of a high-level language.
6. Analyze and identify the functionality of a program encoded in a high level language.
7. Use correctly the basic structures of high-level programming language: conditionals and iterators.
8. Define and properly use data structures that provides a high-level language.
9. Use properly a top-down design to address complex problems by dividing them into a subset of simpler problems.
10. Evaluate and validate the quality of the solution according to the results.
11. Acquire practical skills in solving real problems in both compiled and interpreted environments.

Competences

University of Lleida strategic competences

- UdL3. Mastering ICT's.

Degree-specific competences

- EPS5. Capacity of abstraction and of critical, logical and mathematical thinking.

Degree-transversal competences

- GEM3. Basic knowledge of the use and programming of computers, operating systems, databases and computer programs with applications in engineering.

Subject contents

1. Introduction to Computer Science

    1.1 Evolution, classification and current status of programmable computers

    1.2 Architecture of a computer and its functional units

        1.2.1 Processor and Memory systems

        1.2.2 Computer architecture and I/O

    1.3 Software in Computer Systems

        1.3.1 The operating system

        1.3.2 Programming Languages and Translators (Compilers and Interpreters)

2. Information representation

    2.2 Numbering Systems

    2.3 Basic Binary Operations

    2.4 Signed numbers representation

    2.5 I/O standard codes

3. Introduction to programming in C Language

3.1 Algorithmic Problem Resolution

3.1.1 Programs and Instructions

3.1.2 Algorithmic resolution of a computer problem

3.2 Introduction to the C language

3.2.1 Basic structure of a program

3.2.2 Constants, Variables, Data Types and Operators

3.3. Algorithmic Basic Structures

3.3.1 Sequential Structure.

3.3.2 Conditionals and Selection

3.3.3 Iterative Structures

3.4 Libraries basic functions of C.

3.5 Standard data structures

3.5.1 One dimensional arrays.

3.5.2 Multidimensional Arrays. Matrices

3.5.3 Application of arrays and matrices

4. Introducció to MATLAB

4.1 Interface and basic syntax of MATLAB Language programming

4.2 Top-down algorithm design: Funcion creation and management

4.3 Creating Scripts

4.4 Toolboxes MATLAB

Methodology

This is a mainly practical course in which the study is based on the resolution of recommended and compulsory exercises. Individual work is essential for acquisition of the skills and the correctly use computer tools that will be used during the course.

You can find the following collections of teaching materials in the Virtual Campus: http://cv.udl.cat

1. Introduction to the Computer Basics
2. Introduction to Programming in C
3. MATLAB notes
4. Collections of problems, Examples, Solutions and Bibliography

The use of the Virtual Campus is essential to access the resources of the course, notifications, deadlines information, teaching plan and assessment.

Development plan

Dates (Weeks)	Description	Classroom Activities	FH (2) (Hours)	Autonomous Activity	AH (3) (Hours)
Week 1	Course presentation (GG)	Exhibition events and methodology	1	Teaching Plan Review	2
	T1. Evolution, classification and current status of programmable computers	Participatory classes and classroom activities	1	Study	2
Week 2	T1. Processor and Memory systems	Participatory classes	1	Study	2
	T2. Signed numbers representation. Basic binary operation (GM)	Solving exercises and participatory classes	2	Exercise resolution and study	2
Week 3	T1. Processor and Memory systems T1. Programming languages	Participatory classes ?	2	Study	2
	T2. Signed numbers representation (GM)	Solving exercises and participatory classes	2	Exercise resolution and study	2

Week 4	T3. Algorithmic Problem Resolution (Activity)	Solving probelms and participatory classes	2	Study	2
	T2. Signed numbers representation. Standard I/O codes (GM)	Solving exercises and participatory classes	2	Exercise resolution and study	4
Week 5	T3. Basic structure of a program in C. Variables	Solving probelms and participatory classes	2	Study	2
	PR1. Programming environment and compilation (GM)	Classroom practises	2	Exercise resolution and study	4
Week 6	T3. Conditional structures	Solving problems and participatory classes	2	Study	4
	Programming problemes (GM)	Classroom practises and problem solving	2	Exercise resolution and study	4
Week 7	T3. Iterative structures	Solving problems and participatory classes	2	Study	4
	Problems. Conditional	Classroom practises and problem solving	2	Exercise resolution and study	4
Week 8	T3. Libraries basic functions of C	Solving problems and participatory classes	2	Study	2
	Problems. Iterations	Classroom practises and problem solving	2	Exercise resolution and study	4
Week 9	PA1. Evaluation Test 1	Individual writting exam	2	Exercise resolution and study	2
Week 10	Random Numbers + Arrays	Solving problems and participatory classes	2	Study	2
	PRA1. Evaluation Iterators	Classroom practises and problem solving	2	Exercise resolution and study	4
Week 11	Array exercises Matrix theory	Solving problems and participatory classes	2	Study	4
	Array problems	Classroom practises and problem solving	2	Exercise resolution and study	4
Week 12	Matrix exercises	Solving problems and participatory classes	2	Study	4
	PRA2. Evaluable activity. Arrays + Random numbers	Classroom practises and problem solving	2	Exercise resolution and study	4
Week 13	MATLAB I	Solving problems and participatory classes	2	Study	4
	Matrix Problems	Classroom practises and problem solving	2	Exercise resolution and study	4
Week 14	MATLAB II	Solving problems and participatory classes	2	Study	4
	Practice Matlab I	Classroom practises	2	Exercise resolution and study	4
Week 15	MATLAB III	Solving problems and participatory classes	2	Study	4
	Practice Matlab II	Classroom practises	2	Exercise resolution and study	4
Week 16 i 17	PA2. Evaluation Test 2	Individual writting exam	2	Exercise resolution and study	
Week 19	Recovery Test	Individual writting exam	2	Exam review and study	

(2) FH = Face-to-Face Hours

(3) AH = Autonomous Hours

Evaluation

Objectives	Evaluation activities	%	Dates	O/V (1)	I/G (2)	Remarks
Continous assesment Topic 3	Evaluable practice PRA1	15	Week 10	O	I	
Continous assesment Topic 3	Evaluable practice PRA2	15	Week 12	O	I	

Objectives	Evaluation activities	%	Dates	O/V (1)	I/G (2)	Remarks
Continous assesment Topics 1, 2, 3 i 4	Autonomous work and classroom participation. AP	10	During all the course	O	I	
Continous assesment Topics 1 i 2 i 3	PA1. Writting exam	30	Week 9	O	I	
Continous assesment Topics 3 i 4	PA2. Writting exam	30	Week 16 i 17	O	I	
Recovery assesment. Topics 1, 2, 3 i 4	Recovery assesment. Writting exam (*)	60	Week 19	V	I	Recoveries Week

(1) Mandatory / Voluntary
(2) Individual / Group

Final Marks:
In the weeks 9th and 16th / 17th will be performed the writting exams: PA1 and PA2. Each of these tests has a weight of 30% over the final mark.

The final grade is calculated by the sum of the results of the writting test (PA1, PA2) plus the classroom practices (PRA1, PRA2) and autonomous work and classroom participation (AP): **Final Mark = PA1 + PA2 + PRA1 + PRA2 + AP.** The practices (PRA1, PRA2) and participation (AP) marks will be added to the final mark only when the mark for the written tests (PA1 + PA2) is equal to or greater than 3 over 6.

(*) On the 19th week the grade of the course can be recovered with a final writting exam, following the guidelines of the EPS Academic Degrees Framework. The recovery will be done through a written exam with all contents of the course with a weight of 60% of the final grade. The continuous assessment practices can not be recovered. This test can also be done by all those students having passed the ongoing assessment and want to modify their grade, assuming that the obtained score in the recovery test is the only score that will be considered for the calculation of the final mark.

Bibliography

Basic bibliography:

• DEITEL,HARVEY M. and DEITEL, PAUL J. Como Programar en C/C++. Pearson/Prentice Hall,cuarta edición, 2003.

• GARCÍA,FÉLIX, CARRETERO, JESÚS, FERNÁNDEZ, JAVIER i CALDERÓN, ALEJANDRO. El lenguaje de Programación C.Diseño e implementación de programas. EditorialPrentice Hall, 2002.

• JOYANES,LUIS i ZAHONERO, IGNACIO. Programación en C. Metodología, estructura de datos yobjetos. Editorial Mc Graw-Hill, 2001.

• KERNIGHAN,BRIAN W. and RITCHIE, DENNIS M. The C Programming Language. Editorial PrenticeHall, 2nd edition, 1988.

• WAITE,MITCHELL i PRATA, STEPHEN. Programación en C. Editorial Anaya, tercera edición,1990.

Complementary bibliography:

• HERNÁNDEZ,ENRIQUE, HERNÁNDEZ, JOSÉ i JUAN, MARI CARMEN. C++ estándar. EditorialParaninfo, 2002.

• LLANOS,DIEGO R. Curso de C bajo Unix. Editorial Paraninfo, 2001.

• TANEMBAUN,ANDREW S. Structured Computer Organization. Prentice Hall, Fourth Edition,1999.

• W.STALLINGS,WILLIAM. Organización y Arquitectura de Computadores. Prentice Hall, 5ªEdición, 2000.