Programming

Academic Year: (2016 / 2017) **Review date: 29/04/2016 23:04:45**

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: SESMERO LORENTE, MARIA PAZ

Type: Basic Core ECTS Credits : 6.0

Year : 1 Semester :

Branch of knowledge: Engineering and Architecture

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

There are no pre-requisites for this course

OBJECTIVES

Generic competences and skills
1.Analysis and synthesis skills
2. Organization an planning skills
3. Ability to apply academic knowledge to practice
4. Basic computer skills
5. Ability to proactively solve problems with critical thinking
6. Basic teamwork skills

Specific skills
Cognitive

1. To understand the role of Computing Science and Programming in the context of Industrial Engineering
2. To explain the fundamentals of computer hardware and computer architecture and to identify the basic computer components involved in the execution of statements and programs
3. To describe the concepts of algorithm, program, statement and programming language
4. To explain the fundamentals of computer software: operating systems, types of programs, compilers
5. To understand the concepts of variable, constant, operator and expression, as well as the different operators available in the C programming language
6. To know the basic algorithmic structures: sequential, alternative, iterative
7. To understand the concept of function and the use of parameters
8. To understand the principles of modular and structured programming, as well as the concepts of data and function abstraction
9. To understand the concept of structured data type and use arrays, structures and strings
10. To know the basic search and sort algorithms
11. To know the syntax of the C programming language
12. To know the main programs used to solve Engineering problems

Procedural
1. To solve Engineering problems of medium complexity by designing algorithms and developing computer programs in the C language
2. To select the most appropriate data structures to solve a problem
3. To use an integrated development environment to write, compile and debug programs
4. To use functions of the C standard library

DESCRIPTION OF CONTENTS: PROGRAMME

Topic 1. Introduction to Computer Science and Programming.
1.1. Computers and Computer Science. Brief history of computers

1.2. Information representation with computers
1.3. Algorithms and programs. Tools for algorithm design
1.4. Computer Science in the context of Industrial Engineering

Topic 2. Software and Hardware
2.1.Logic support: Software.
2.1.1. Programming languages.
2.1.2. Translators, compilers and interpreters.
2.1.3. Operating systems.
2.2. Physical support: Hardware.
2.2.1. Computer architecture. Components.
2.2.2. Program execution.
2.2.3. Peripherals.
2.2.4. Computer networks and the Internet

Topic 3. Introduction to Programming in C
3.1. Basic program structure
3.2. Variables and constants
3.3. Simple data types
3.4. Expressions and instructions
3.5. Operators: arithmetic, relational, logical and assignment operators
3.6. Pointer type
3.7. Basic input and output

Topic 4. Control Flow and Loops
4.1. Conditional instructions:
4.1.1. if-else
4.1.2. switch
4.2.Loop instructions:
4.2.1. for
4.2.2. while
4.2.3. do-while
4.3.Nested control flow and loops

Topic 5. Functions
5.1. Modular programming
5.2. Function declaration and definition
5.3. Function calling
5.4. Parameters ¿ call by value and by reference
5.5. Scope of variables in functions
5.6. Library functions and standard C libraries

Topic 6. Complex Data Types
6.1. Introduction: structured vs. simple data types
6.2. Arrays.
6.2.1. Arrays: definition and use
- 6.2.2. Arrays and pointers
6.2.3. Character strings
6.2.4. Arrays as function parameters
6.3. Structures
6.3.1. Structures: definition and use
6.3.2. Arrays of structures
6.3.2. Structures as function parameters

Topic 7. Search, Sort and Merge Algorithms
7.1. Search algorithms
7.2. Sort algorithms
7.3. Merge algorithms

Topic 8. Advanced topics
8.1. External data storage: files and databases
8.2. Dynamic memory management

8.3. Programs used in Engineering

LEARNING ACTIVITIES AND METHODOLOGY

Lectures
Lectures will be developed in joint student groups. Professors will explain the contents of the course to support the students to acquire the related cognitive skills.

Exercise sessions
Exercise lectures will be developed in small groups. Professors and students will solve problems to acquire the procedural skills of the course.

Computer lab sessions
Computer labs will be developed. Students will implement and test problem solutions on a computer.

Individual tutorships
Students can request for individual sessions with the professors to ask specific questions about the course contents and exercises.

Individual work
Students will study the contents of the course presented in the lectures and solve the exercises proposed in exercise and computer lab sessions. They will receive feedback from the professors to detect and correct their errors.¿

ASSESSMENT SYSTEM

% end-of-term-examination/test:	60
% of continuous assessment (assigments, laboratory, practicals…):	40

Continuous asessment: 40 % of the final mark

Three parts

Test 1:
Content: Test on lessons 3 and 4 (Introduction to Programming in C, Control flow and Loops)
Value: 10% of the final grade

Exercise 1. Programming problem on lessons 3, 4, 5 and partially 6 (Introduction to Programming in C, Control flow and Loops, Functions, Complex Data Types-only arrays)
Value: 20% of the final grade

Exercise 2. Programming problem on lessons 3, 4, 5 y 6. This exam takes place in the computing lab.
Value: 10% of the final grade

Final exam
Value: 60% of the final grade
Two parts:
- Test, which will cover all topics of the course. Multiple choice test, four answers, only one correct. Penalties for wrong answers will be applied (1/3 of a right answer).
- Two problems, which will require the students to design and implement a C program.

June
The exam will have the same structure as the May exam.
Continuous evaluation will not be taken into account in this exam

Please note that to pass this course, a mark above 4 (out of 10) in the final exam is required

BASIC BIBLIOGRAPHY

- Brian W. Kernighan, Dennis M. Ritchie C Programming Language, Prentice Hall, 1988 (2nd Edition)

- K. N. King. C Programming: A Modern Approach. , W.W. Norton & Company, , 2008 (2nd Edition)

- Paul J. Deitel, Harvey M. Deitel. C: How to Program. , Prentice Hall, (6th Edition), 2009