Computing Systems I

Academic Year: (2019 / 2020) **Review date: 18/12/2019 18:45:12**

Department assigned to the subject: Systems Engineering and Automation Department

Coordinating teacher: ABDERRAHIM FICHOUCHE, MOHAMED

Type: Compulsory ECTS Credits : 6.0

Year : 3 Semester : 1

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Students are expected to have knowledge about basic programming concepts and computer systems such as those covered in the subject Programming of the first year of the engineering program.

OBJECTIVES

By the end of this subject, students will be able to have:
1. a systematic understanding of the key aspects and concepts of their branch of engineering in computing systems;
2. coherent knowledge of their branch of engineering including some at the forefront of the branch in computing systems;
3. the ability to apply their knowledge and understanding of computing systems to identify, formulate and solve engineering problems using established methods;
4. the ability to apply their knowledge and understanding to develop and realise designs to meet defined and specified requirements;
5. an understanding of design methodologies, and an ability to use them.
6. workshop and laboratory skills.
7. the ability to select and use appropriate equipment, tools and methods;
8. the ability to combine theory and practice to solve problems of computing systems;
9. an understanding of applicable techniques and methods in computing systems, and of their limitations;

DESCRIPTION OF CONTENTS: PROGRAMME

1. Introduction to computer systems development

2. Requirements Engineering
2.1 Introduction to Requirements Engineering
2.2 what are Requirements
2.3 Classification of Software Requirements

3. Unified Modeling Language UML and Object Oriented Analysis
3.1 Structural Modeling
3.1.1 General Introduction to all structural diagrams
3.1.2 Classes Diagram (detailed)
3.2 Dynamical modeling
3.2.1 General Introduction to all Dynamic diagrams
3.2.2 Activity Diagrams
3.2.3 Sequence Diagrams
3.2.4 State Machine Diagrams
3.2.5 Use Cases Diagrams

4. Object-Oriented Design
4.1 DOO Concepts
4.2 DOO Methods

5. Introduction to Object Oriented Programming in C + +.
5.1 Syntax of C + +.
5.2 Basic Programming
5.3 Introduction to classes, objects and methods.
5.4 Class hierarchy and inheritance
5.5 Polymorphism.
5.6 C++ Templates
5.7 Input and Output Management (files)
5.8 Exception Handling

LEARNING ACTIVITIES AND METHODOLOGY

The learning activities include:

Lectures, classes for resolution of doubts in small groups, student presentations, tutorials and individual work of students; aimed at the acquisition of knowledge (3 ECTS).
Laboratory practices and sections of problems in small groups, individual tutorials and individual work of students, aimed at the acquisition of practical skills related to the syllabus of the subject (3 ECTS).

ASSESSMENT SYSTEM

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

The evaluation system includes continuous assessment of student work (papers, reports of laboratory practice, class participation and skills assessment tests of theoretical and practical knowledge) and the final assessment through a written final exam in which the knowledge, skills and abilities acquired throughout the course will be evaluated comprehensively . The percentages allocated is: 60% continuous assessment and 40% final exam.

BASIC BIBLIOGRAPHY

- Bjarne Stroustrup The C++ Programming Language, fourth Edition, Addison-Wesley, 2013

- Harvey. M. Deitel and Paul. J. Deitel C++ How to Program ¿ (9th Edition: introducing the new C++11 Standard), Prentice Hall, 2011

- Joseph Schmuller Sams Teach Yourself UML in 24 Hours, Third Edition, Sams Publishing, 2004

- Stanley B. Lippman, Josée Lajoie "C++ Primer",Tercera Edición,, Addison-Wesley,, 1998