

Academic Year: (2021 / 2022)

Review date: 04-06-2021

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: ONORATI , TERESA

Type: Compulsory ECTS Credits : 6.0

Year : 4 Semester : 1

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Programming (Course: 1 / Semester: 1)

Principles of Computer Engineering (Course: 1 / Semester: 2)

Automata and Formal Language Theory (Course: 2 / Semester: 1)

OBJECTIVES

Learning outcomes:

R1 - Knowledge and understanding: have basic knowledge and understanding of the scientific and technological foundations of Computer Engineering, as well as specific knowledge of computer science, computer engineering, and information systems.

R2 - Engineering Analysis: Be able to identify Computer Engineering problems, recognize their specifications, establish different resolution methods and select the most appropriate one for their solution, considering the social, human health, environmental, and commercial constraints applicable in each case.

R3 - Engineering Design: Be able to perform engineering designs according to their knowledge and understanding that meet the required specifications in collaboration with other engineers and graduates. Design encompasses devices, processes, methods and objects, and specifications broader than strictly technical, including social awareness, health, and safety, environmental and commercial considerations.

R4 - Research and Innovation: Be able to use appropriate methods to conduct research and make innovative contributions in Computer Engineering.

R5 - Engineering Applications: Graduates will be able to apply their knowledge and understanding to solve problems, conduct research, and design devices or processes in the field of Computer Engineering according to criteria of cost, quality, safety, efficiency, environmental friendliness, and ethical implications. These skills include knowledge, use and limitations of computer systems, process engineering, computer architectures, computational models, equipment, practical work, technical literature and information sources.

Basic and general competences:

CG2 - Be able to generate new ideas (creativity) and to anticipate new situations and to adapt to work in a team and interact with others, but at the same time have the ability to work autonomously.

CG1 - Apply appropriate theoretical and practical methods for analysis, design, and problem solving, providing computer solutions that respect the rules of accessibility, ergonomics, and safety at work and comply with existing legislation.

CG3 - To evaluate the different possible solutions from the technical, economic, and professional points of view and respect the legislation in force in the general and professional field.

CGB3 - Ability to understand and master the basic concepts of discrete mathematics, logic, algorithmic and computational complexity, and their application to the resolution of engineering problems.

CGB4 - Basic knowledge of the use and programming of computers, operating systems, databases, and computer programs with application in engineering.

CGB5 - Knowledge of the structure, organization, operation, and interconnection of computer systems, the fundamentals of their programming, and their application to the resolution of engineering problems.

CG9 - Efficient use of ICT means writing technical reports and project reports and works on computer science and quality presentations.

CGO3 - Ability to design, develop, evaluate and ensure the accessibility, ergonomics, usability, and security of computer systems, services, and applications, as well as the information they manage.

CGO8 - Knowledge of basic subjects and technologies that enable them to learn and develop new methods and technologies and those that provide them with great versatility to adapt to new situations.

CGO9 - Ability to solve problems with initiative, decision-making, autonomy, and creativity. Ability to know how to communicate and transmit the knowledge, skills, and abilities of the profession of Technical Engineer in Computer Science.

CB2 - That students know how to apply their knowledge to their work or vocation professionally and possess the skills that are usually demonstrated through the development and defense of arguments and problem solving within their area of study.

DESCRIPTION OF CONTENTS: PROGRAMME

1. Introduction to HCI
 - 1.1. What is HCI, why is it needed, what is its relationship to interface design, and how does it relate to HCI?
 - 1.2. History of HCI
 - 1.3. Some examples from everyday life
2. User Interfaces
 - 2.1. What is a user interface and what is it for?
 - 2.2. User Centred Design: Usability; Principles, Guidelines, Heuristics and Patterns; Design Methodology; Prototyping
 - 2.3. Design for All (Universal Design)
3. Web User Interfaces
 - 3.1. What is the Web? History and Evolution of the WWW
 - 3.2. Structure and Navigation of a Web Site
 - 3.3. Web design principles, Heuristics and Patterns
4. Interaction with User Interfaces
 - 4.1. User Experience (UX) design
 - 4.2. Web page interaction design (agile, flat, responsive,...)
 - 4.3. Predictive models: Fitt's Law; Sterring's Law
 - 4.4. Descriptive models: KLM; GOMS
 - 4.5. Inspection methods
 - 4.6. Interaction Paradigms: Large Scale Computing; Personal Computing; Mobile Computing; Ubiquitous Computing; Network Computing; Reality Computing (Augmented Reality and Virtual Reality)

LEARNING ACTIVITIES AND METHODOLOGY

- * Lectures: 2 ECTS. They aim to achieve the specific cognitive competencies of the subject and the transversal competencies of analysis and abstraction.
- * Practical classes: 1 ECTS. They aim to develop the specific instrumental competencies and the transversal competencies problem solving and application of knowledge.
- * Programming exercises: 0.75 ECTS. Initiated during the practical classes and completed outside of them, they aim to develop the specific instrumental competencies, start developing the specific attitudinal competencies, and transversal competencies problem solving and application of knowledge.
- * Case study: 1,75 ECTS. Started during the practical classes and completed outside of them, it aims to complete and integrate the development of all specific and transversal competencies with the design and implementation of a case study through group work.
- * Tutorials: TUTORIALS. Individual or group tutoring sessions organized by the teacher for the students.
- * Final exam: 0,5 ECTS. It aims to influence and complement the development of specific cognitive and procedural skills. It reflects especially the use of the lectures.

ASSESSMENT SYSTEM

The evaluation system includes the assessment of guided academic activities and practical cases, with the following weights:

- * Programming exercises: 20%
Students must submit two different exercises, each one of them represents ten percent (10%) of the final grade.
- * Case study: 40%
Students must submit two different exercises. The first one about prototyping represents ten percent (10%) of the final grade. The second one about implementation and documentation represents thirty percent (30%) of the final grade.
- * Final exam: 40%

The final exam is mandatory and the final mark must be higher than 5 of 10.

% end-of-term-examination:	40
% of continuous assessment (assignments, laboratory, practicals...):	60

BASIC BIBLIOGRAPHY

- Dix, A., Finlay, J., Abowd, G., Beale, R. Human-Computer Interaction, Prentice Hall, 3rd Edition, 2004, 2004
- Nielsen, J. Designing Web Usability, New Riders, 2000
- Preece, J. Interaction Design. Beyond human computer interaction., John Wiley & Sons, 2002
- Shneiderman, B. Designing the User Interface., Addison-Wesley, 3rd Edition, 1999

ADDITIONAL BIBLIOGRAPHY

- Ballard, B. Designing the mobile user experience., Willey, 2007
- Basham, B., Sierra, K. & Bates, B. Head First Servlets and JSP: Passing the Sun Certified Web Component Developer Exam., O'Really Media, 2008.
- Castro, E. HTML, XHTML and CSS., Peachpit Press, 2006.
- Cooper, A.m Reinmann, R., Cronid, D. About Face 3: The Essentials of Interaction Design., Wiley, 2007.
- Flanagan, D. JavaScript: The Definitive Guide., O'Really Media, 2006.