

Academic Year: (2019 / 2020)

Review date: 20-04-2020

Department assigned to the subject: Computer Science and Engineering Department

Coordinating teacher: GARCIA HERRERO, JESUS

Type: Compulsory ECTS Credits : 6.0

Year : 4 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Programming
Automata and Formal Language Theory

OBJECTIVES

General competences:

- Capacity of analysis and synthesis (PO a)
- Capacity to organize and plan (PO c, e)
- Problem solving (PO c)
- Teamwork (PO d)
- Capacity to apply theoretical concepts (PO a, c)

Specific competences

- Cognitive
 1. Knowledge on theoretical basis of automata and formal languages (PO a)
 2. Knowledge on techniques for lexical, syntactic and semantic analysis (PO a)
 3. Techniques for code generation (PO c)
 4. Techniques for error recovery (PO c)
 5. Knowledge on code optimization methods (PO a, c)
- Procedimental/Instrumental
 1. Design of a formal grammar (PO c)
 2. Design of a lexical and syntactic analyzers (PO c)
 3. Use of automatic tools (meta-compilers) for generation of analyzers (PO k)
- Attitudinal (PO a, c, d)
 1. Ability to generate new ideas (creativity)
 2. Concern with quality
 3. Motivation for success
 4. Interest for investigating and finding solutions to new problems

DESCRIPTION OF CONTENTS: PROGRAMME

Representation of formal languages, lexical analysis, syntactic analysis, semantic analysis, code generation, error recovery, code optimization

LEARNING ACTIVITIES AND METHODOLOGY

Theoretical lectures: 1.5 ECTS. To achieve the specific cognitive competences of the course (PO a, c).

Practical lectures: 1,5 ECTS. To develop the specific instrumental competences and most of the general competences, such as analysis, abstraction, problem solving and capacity to apply theoretical concepts. Besides, to develop the specific attitudinal competences. They consist in proposing during the practical lectures a compiler/interpreter project to be developed in teamwork (PO c, d, e, g, k).

-Guided academic activities (present teacher): 1 ECTS. The student proposes a project according to the teachers guidance to go deeply into some aspect of the course, followed by public presentation (PO c, d, g).

-Guided academic activities (absent teacher): 1.5 ECTS. Exercises and complementary readings proposed by teacher (PO a, c).

Exercises and examination: 0,5 ECTS. To complete the development of specific cognitive and procedimental capacities (PO a, c).

ASSESSMENT SYSTEM

Exercises and examinations are both learning and evaluation activities. The evaluation system includes the assessment of guided academic activities and practical cases, with the following weights:

Exercises and examination: 40% (PO a, c)

Practical case: 40% (PO c, d, e, g, k)

Guided academic activities

- Present teacher: 15% (PO a, c)

- Absent teacher: 5% (PO a, c, k)

% end-of-term-examination: 40

% of continuous assessment (assignments, laboratory, practicals...): 60

BASIC BIBLIOGRAPHY

- A. V. Aho and Ravi Sethi and J. D. Ullman Compiladores: Principios, Técnicas y Herramientas, Addison-Wesley Iberoamericana, 1990.

- Kenneth C. Loudon Construcción de Compiladores. Principios y práctica, Thomson, 2004.

ADDITIONAL BIBLIOGRAPHY

- C. N. Fisher, R. J. Leblanc Crafting a Compiler with C, Addison-Wesley, 1991.

- Dick Grune, Henri E. Bal, Criel J.H. Jacobs, Koen G. Langendoen Modern Compiler Design, John Wiley & Sons, 2000.

- Doug Brown, John Levine, Tony Mason Lex & Yacc, O'Reilly Media, Inc., 1995.

- F. J. Sanchis and C. Galán Compiladores: Teoría y Construcción, Paraninfo, 1986.

- Garrido, Iñesta, Moreno, Pérez Diseño de Compiladores, Publicaciones Universidad de Alicante, 2002.

- K. A. Lemone Fundamentals of Compilers: An Introduction to Computer Language Translation, CRC Press, 1992.