1. OVERVIEW

Subject area	Compilers and Formal Languages
Degree	Bachelor's Degree in Computer Engineering
School/Faculty	Architecture, Engineering and Design
Year	Fourth
ECTS	6 ECTS
Type	Compulsory
Language(s)	Spanish
Delivery mode	On campus / Online
Semester	Second
Year	2022/2023
Coordinating professor	On-campus: Professor Leopoldo Santos Santos
Teacher	On-campus: Professor Leopoldo Santos Santos Online: Professor José Delgado Pérez

2. INTRODUCTION

Compilers and Formal Languages is a compulsory subject within the Bachelor's Degree in Computer Engineering, taught in the 2nd semester of the 4th year. It is worth 6 ECTS credits, like all other compulsory subject areas within the degree programme.

The learning outcomes of this subject area focus on developing concepts related to formal language processing and compiler construction that are necessary to understand how a programming language is designed and implemented. This subject area is essential for learning techniques that allow for the development of word processors, web browsers, systems that use pre-existing language processing or even a program's secure coding techniques.

3. SKILLS AND LEARNING OUTCOMES

Basic skills (CB, by the acronym in Spanish):

CB4. Students can communicate information, ideas, problems and solutions to both specialist and non-specialist audiences.

CB5. Students have developed the learning skills necessary to undertake further study in a much more independent manner.

Transversal skills (CT, by the acronym in Spanish):

- **CT10**. Initiative and entrepreneurial spirit: Ability to undertake difficult or risky actions with resolve. Ability to anticipate problems, propose improvements and persevere to ensure they are implemented. Willingness to take on and carry out tasks.
- **CT14**. Innovation/Creativity: Ability to propose and invent new, original solutions that contribute towards improving problem situations, including ideas from other contexts.
- **CT16**. Decision-making: Ability to choose between different options or methods to effectively solve varied situations or problems.
- **CT18**. Use of information and communication technology (ICT): Ability to effectively use information and communication technology, such as tools for searching, processing and storing information, and for developing communication skills.

General skills (CG, by their acronym in Spanish):

CG3. Ability to design, develop, assess and ensure the accessibility, ergonomics, usability and security of systems, services and computer applications, as well as the information they manage.

Specific skills (CE, by the acronym in Spanish):

CE26. Ability to understand the theoretical foundations of programming languages and its associated lexical, syntactic and semantic processing techniques, and know how to apply them to create, design and process languages.

Learning outcomes (RA, by the acronym in Spanish):

RA1. Explain how the different stages that make up a compiler's analysis phase operate.
RA2. Interpret how the different stages that make up a compiler's synthesis phase operate.
RA3. Use the algorithms that have been learnt to solve specific problems.
RA4. Build a source code to object code translator for given languages.
The following table shows how the skills developed in the subject area match up with the intended learning outcomes:

Skills	Learning outcomes
CB4, CT16, CG3, CE26.	**RA1.** Explain how the different stages that make up a compiler's analysis phase operate.
CB4, CB5, CT10, CT14, CT16, CG3, CE26	**RA2.** Interpret how the different stages that make up a compiler's synthesis phase operate.
CB4, CB5, CT14, CT16, CT18, CG3, CE26	**RA3.** Use the algorithms that have been learnt to solve specific problems.
CB4, CT10, CT16, CT18, CG3, CE26	**RA4.** Build a source code to object code translator for given languages.

Skills	Learning outcomes
CB4, CT16, CG3, CE26.	**RA1.** Explain how the different stages that make up a compiler's analysis phase operate.

CB4, CB5, CT10, CT14, CT16, CG3, CE26	**RA2.** Interpret how the different stages that make up a compiler's synthesis phase operate.
CB4, CB5, CT14, CT16, CT18, CG3, CE26	**RA3.** Use the algorithms that have been learnt to solve specific problems.
CB4, CT10, CT16, CT18, CG3, CE26	**RA4.** Build a source code to object code translator for given languages.

4. CONTENTS

The subject is organised into six learning units, which in turn are divided into topics (four or five topics depending on the units):

Unit 1. Introduction to Compilers and Formal Languages
1.1. Foundations of Compilers.
1.2. Formal Languages
1.3. Turing Machines.

Unit 2. Lexical analysis
2.1. Basic concepts of lexical analysis.
2.2. Regular expressions.
2.3. Finite-state machines.
2.4. From Regular Expression to DFA.

Unit 3. Syntactic analysis I
3.1. Grammar.
3.2. Syntactic analysers.
3.3. FIRST and FOLLOW sets.
3.4. Descending syntactic analysis LL(1).

Unit 4. Syntactic analysis II
4.1. SLR.
4.2. LR1.
4.3. LALR.
4.4. Ascending syntactic analysers generator.

Unit 5. Semantic Analysis
5.1. Attribute grammars and types.
5.2. Symbol table.
5.3. ETDS vs DDS.
5.4. Type checking.

Unit 6. Code generation
6.1. Three-address code
6.2. Examples for different data structures.
6.3. Memory and parameters
6.4 Code optimization

5. TEACHING/LEARNING METHODS

The types of teaching/learning methods are as follows:

- Lecture.
- Laboratory work.
- Problem-solving individually and in groups
- Project-based learning.

6. LEARNING ACTIVITIES

The types of learning activities, plus the amount of time spent on each activity, are as follows:

On campus:

Learning activity (AF, by the acronym in Spanish)	Number of hours
Lectures	50
Integrative group work	25
Independent working	50
Tutorials	25
TOTAL	**150**

Online:

Learning activity (AF, by the acronym in Spanish)	Number of hours
Independent working	55
Reading of content	50
Integrative group work	25
Tutorials and academic monitoring	25
TOTAL	**150**

7. ASSESSMENT

The assessment systems, plus their weighting in the final grade for the subject area, are as follows:

On campus:

Assessment system	Weighting
Exams and tests	30%
Report writing	30%
Alternative assessment methods	25%
Exercises, problems and case studies	15%

Online:

Assessment system	Weighting
Knowledge test	60%
Development of articles, reports or design briefs	20%
Alternative assessment methods with mind maps, diaries, debates, portfolios, peer assessment	20%

On the Campus Virtual, when you open the subject area, you will find all the details of your assessable tasks and the deadlines and assessment procedures for each task.

7.1. Ordinary exam period

The conditions to pass the subject area are as follows:

1. You must achieve a grade of at least 5 in the exam on the subject area.
2. The compiler must work in compliance with the minimum requirements indicated in the activity. The grade for the compiler must be at least 5.
3. The average grade for the activities (including the exam and compiler) must be at least 5.0 out of 10.0.

7.2. Extraordinary exam period (resits)

Activities not passed in the ordinary exam period, or those not submitted, must be submitted after receiving the relevant corrections and feedback from the lecturer.

Once the activities indicated by the professor have been submitted, the conditions to pass the subject area are as follows:

1. You must achieve a grade of at least 5 in the exam on the subject area.
2. The compiler must work in compliance with the minimum requirements indicated in the activity. The grade for the compiler must be at least 5.
3. The average grade for the activities (including the exam and compiler) must be at least 5.0 out of 10.0.

5

8. TIMELINE

The timeline with submission dates for the assessable tasks in this subject area will be indicated in this section:

Assessable tasks	Date
Activity 1. Programming Language Classification	Weeks 1–2
Activity 2. Solving a Regular Expression to DFA conversion exercise	Weeks 3–4
Activity 3. Carrying out a lexical analyser	Weeks 4–5
Activity 4. Solving a complete LL(1) parser exercise	Weeks 6–7
Activity 5. Carrying out an exercise for every type of ascending LR parser	Weeks 8–10
Activity 6a. Using the lexical analyser from activity 3, obtain grammar that allows for language/XML processing.	Weeks 11–13
Activity 6b. Incorporate semantic checks on the outcomes of activity 6a and create code	Weeks 14–15
Activity 7. Exam/Knowledge test	Weeks 18–19

The timeline may be subject to change for logistical reasons related to the activities. Students will be informed of any changes in due time and course.

9. BIBLIOGRAPHY

The reference material for the subject area is as follows:

- Louden K.C. (2004), Construcción de Compiladores: Principios y práctica. Thomson Learning, Mexico ISBN: 970-686-299-4.
- Aho A.V., Lam M. S., Sethi R., Ullman J.D.(2008) Compiladores, principios, técnicas y herramientas (2ª Edición) Perason Educación, México ISBN: 978-970-26-1133-2.
- Other recommended sources are:
  Aho A.V., Sethi R., Ullman J.D. (1986). Comipiladores principios, técnicas y herramientas. Addison- Wesley Publishing Company. Traducción de 1990 por Addison-wesley Iberoamericana, S. A
  Isasi P., Martínez P., Borrajo D. (1997). “Lenguajes, Gramáticas y Autómatas. Un enfoque práctico”, Addison-Wesley.

10. DIVERSITY AWARENESS UNIT

Students with special educational needs:

6

To ensure equal opportunities, curricular adaptations or adjustments for students with special educational needs will be outlined by the Diversity Awareness Unit (UAD, Spanish acronym).

As an essential requirement, students with special educational needs must obtain a report about the curricular adaptations/adjustments from the Diversity Awareness Unit by contacting unidad.diversidad@universidadeuropea.es at the beginning of each semester.

11. STUDENT SATISFACTION SURVEYS

Your opinion matters!
Universidad Europea encourages you to complete our satisfaction surveys to identify strengths and areas for improvement for staff, degrees and the learning process.
These surveys will be available in the survey area of your campus virtual or by email.
Your opinion is essential to improve the quality of the degree.
Many thanks for taking part.