

## SUBJECT TEACHING GUIDE

### G1662 - Programming Languages

#### Degree in Computer Systems Engineering

Academic year 2024-2025

| 1. IDENTIFYING DATA              |   |                  |                    |                  |                  |
|----------------------------------|---|------------------|--------------------|------------------|------------------|
| Degree                           | Degree in Computer Systems Engineering  |                  |                    | Type and Year    | Optional. Year 4 |
| Faculty                          | Faculty of Sciences   |                  |                    |                  |                  |
| Discipline                       | Subject Area: Computing<br>Mention in Computing   |                  |                    |                  |                  |
| Course unit title and code       | G1662 - Programming Languages   |                  |                    |                  |                  |
| Number of ECTS credits allocated | 6   | Term             | Semester based (2) |                  |                  |
| Web                              | <a href="https://moodle.unican.es/course/view.php?id=12160">https://moodle.unican.es/course/view.php?id=12160</a> |                  |                    |                  |                  |
| Language of instruction          | Spanish   | English Friendly | No                 | Mode of delivery | Face-to-face     |

|                  |  |  |  |  |  |
|------------------|--|--|--|--|--|
| Department       | DPTO. MATEMATICAS, ESTADISTICA Y COMPUTACION                           |  |  |  |  |
| Name of lecturer | DOMINGO GOMEZ PEREZ  |  |  |  |  |
| E-mail           | domingo.gomez@unican.es  |  |  |  |  |
| Office           | Facultad de Ciencias. Planta: + 3. DESPACHO DOMINGO GOMEZ PEREZ (3005) |  |  |  |  |
| Other lecturers  |  |  |  |  |  |

| 3.1 LEARNING OUTCOMES  |
|--|
| - Practical knowledge of development of compilers for domain specific languages.                                       |
| - Hands-on experience on back ends and front-ends. The use of Scoping and declarative programming in modern compilers. |
| - Knowledge of applicability of compilers, as in web browsers.   |

#### 4. OBJECTIVES

Apply the skills acquired in formal languages for compiler development.

Understand the intrinsic complexity to generate machine code

Understand the use of pushdown automata and their role in a compiler.

Generate concise error messages for source code.

Learn the abilities necessary to use of parser generator tools.

#### 6. SUBJECT PROGRAM

##### CONTENTS

|     |   |
|-----|---|
| 1   | Part 1: Theoretical background              |
| 1.1 | Background and overview of formal languages |
| 1.2 | Bottom up parsing                           |
| 1.3 | Types of bottom up parsing                  |
| 1.4 | Attribute grammars                          |
| 1.5 | Type checking and code generation           |
| 1.6 | Bootstrapping                               |
| 2   | Part 2: Tools                               |
| 2.1 | Python                                      |
| 2.2 | Lexer generation                            |
| 2.3 | Parser generators                           |
| 2.4 | Intermediate code                           |
| 2.5 | Native code                                 |

| 7. ASSESSMENT METHODS AND CRITERIA  |  |             |           |        |
|---|--|-------------|-----------|--------|
| Description   | Type                                   | Final Eval. | Reassessn | %      |
| Individual assignment   | Work                                   | No          | Yes       | 30,00  |
| Group assignment  | Work                                   | No          | Yes       | 30,00  |
| Classroom assignment  | Activity evaluation with Virtual Media | No          | Yes       | 10,00  |
| Practical exam  | Written exam                           | Yes         | Yes       | 30,00  |
| TOTAL   |  |             |           | 100,00 |
| Observations  |  |             |           |        |
| <p>Students can substitute continuous evaluation by a practical exam if:</p> <ul style="list-style-type: none"> <li>- they are registered as a part-time student,</li> <li>- they fail any of the task in the continuous evaluation or want to improve their grades.</li> </ul> <p>The value equals to 50% of the final grade and it lasts for approximately 2 hours. It will be required that the student presents all pending assignments to apply to this condition.</p> <p>In the case that socio-sanitary conditions advise the end of in-person teaching, the final exam will be replaced by a practical exam and a written exam. The availability of resources will determine how the exams are performed, attempting to implement a asynchronous and telematic method.</p> <p>In the case that socio-sanitary conditions advise the end of in-person teaching, the final exam will be replaced by a practical exam and a written exam. The availability of resources will determine how the exams are performed, attempting to implement a asynchronous and telematic method.</p> |  |             |           |        |
| Observations for part-time students   |  |             |           |        |
| <p>Any student who is registered as partial-time student is allowed to choose between continuous evaluation or two final exams. The percentage of the exams in the final grade will be 50% and 50%, respectively.</p>   |  |             |           |        |

## 8. BIBLIOGRAPHY AND TEACHING MATERIALS

### BASIC

Torben Mogensen. Basics of compiler design

Alfred Aho, Monica Lam, Ravi Sethi, and Jeffrey Ullman.

Compilers: Principles, Techniques, and Tools (Second Edition)