

Course guide

270025 - LP - Programming Languages

Last modified: 30/01/2024

Unit in charge: Barcelona School of Informatics
Teaching unit: 723 - CS - Department of Computer Science.

Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Optional subject).

Academic year: 2023 **ECTS Credits:** 6.0 **Languages:** Catalan

LECTURER

Coordinating lecturer: JORDI PETIT SILVESTRE

Others:

Primer quadrimestre:
GERARD ESCUDERO BAKX - 12, 21, 22
ANA EDELMIRA PASARELLA SANCHEZ - 11
JORDI PETIT SILVESTRE - 11, 12, 21

Segon quadrimestre:
JORDI DELGADO PIN - 11
GERARD ESCUDERO BAKX - 11, 12, 13, 21
ANA EDELMIRA PASARELLA SANCHEZ - 13, 22
JORDI PETIT SILVESTRE - 21, 22

PRIOR SKILLS

Prior skills on logic acquired in the course on Foundations of Mathematics (FM):

- Knowledge of basic concepts of logic propositions and predicates
- Knowledge of logical inference.

Prior programming skills acquired in the courses on Data structures and algorithms (EDA) and Programming Projects (PROP):

- Knowledge of programming.
- Knowledge of object oriented programming languages

REQUIREMENTS

- Corequisite PROP

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:

CCO1.2. To demonstrate knowledge about the theoretical fundamentals of programming languages and the associated lexical, syntactical and semantic processing techniques and be able to apply them to create, design and process languages.

Generical:

G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

TEACHING METHODOLOGY

Classes are divided into theory and lab sessions.

In the theory sessions the concepts of the subject will be developed. The presentation of new theoretical material is combined with the resolution of examples and the interaction with the students to promote the discussion of introduced concepts.

In the laboratory classes the introduced concepts will be put into practice and applied to specific problems and programming languages. There will be three small practices associated with the different evaluation tests.

LEARNING OBJECTIVES OF THE SUBJECT

1. Knowing the different compilation steps, including lexical, syntactic and semantic.
2. Conocer herramientas para la creación de analizadores léxicos y sintácticos.
3. Knowing the differences between a compiler and an interpreter.
4. Distinguish the main features of a given programming language. Identify its strengths and weaknesses and be able to justify a choice.
5. Understand the main features of functional programming languages.
6. Learn advanced programming language concepts
7. Being able to model and specify hardware or software problems using functional languages.
8. Learn the main features of scripting languages.
9. Ability to learn new programming languages independently.

STUDY LOAD

Type	Hours	Percentage
Guided activities	6,0	4.00
Hours small group	30,0	20.00
Self study	84,0	56.00
Hours large group	30,0	20.00

Total learning time: 150 h

CONTENTS

Introduction to programming languages.

Description:

Turing completeness. Programming paradigms. Main features.

Introduction to compilers.

Description:

Compilers vs interpreters. The lexical analysis. Parsing. Grammars. Table of symbols. Abstract syntax tree. Semantic analysis. Code generation.

Functional languages.

Description:

Foundations. Pattern Matching. Eager/Lazy evaluation. Introduction to Haskell.



Type systems.

Description:

Types in programming languages□□. Types and reliability. Type systems in functional languages □□and object-oriented languages□□.

Higher-order programming

Description:

Higher-order functions. Applications. Programming higher-order functions in functional and object-oriented languages□□.

Modeling and Specification using functional languages

Description:

Modeling and specification using declarative languages □□. Prototyping. Executable specifications. Program transformation.

Scripting languages

Description:

Features. Using scripting languages□□. Combining paradigms. Types. Higher-order features. Short introduction to Python.

ACTIVITIES

Introduction to programming languages

Specific objectives:

4, 5, 9

Related competencies :

G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 6h

Theory classes: 2h

Self study: 4h

Introduction to compilers

Specific objectives:

1, 2, 3

Full-or-part-time: 18h

Theory classes: 2h

Laboratory classes: 8h

Self study: 8h

CL

Specific objectives:

1, 2, 3

Full-or-part-time: 10h

Guided activities: 2h

Self study: 8h

Functional languages

Specific objectives:

4, 5, 9

Related competencies :

G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 18h

Theory classes: 4h

Laboratory classes: 6h

Self study: 8h

Type systems

Specific objectives:

4, 5, 6, 9

Related competencies :

G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 10h

Theory classes: 4h

Self study: 6h

Higher-order programming

Specific objectives:

4, 5, 6, 9

Related competencies :

G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 10h

Theory classes: 2h

Laboratory classes: 2h

Self study: 6h

Modeling and specification using functional languages

Specific objectives:

5, 7

Related competencies :

G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 12h

Theory classes: 4h

Laboratory classes: 2h

Self study: 6h

EP

Specific objectives:

5, 6

Full-or-part-time: 12h

Guided activities: 2h

Self study: 10h

Scripting languages

Specific objectives:

3, 4, 8, 9

Related competencies :

G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 14h

Theory classes: 2h

Laboratory classes: 6h

Self study: 6h

Learning a new programming language

Full-or-part-time: 8h

Theory classes: 2h

Laboratory classes: 2h

Self study: 4h

CT

Specific objectives:

4, 9

Related competencies :

G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 7h

Guided activities: 1h

Self study: 6h

Reviewing activities

Full-or-part-time: 10h

Theory classes: 2h

Laboratory classes: 2h

Self study: 6h

EF

Specific objectives:

4, 5, 6, 7, 8

Related competencies :

G4. EFFECTIVE ORAL AND WRITTEN communication: To communicate with other people knowledge, procedures, results and ideas orally and in a written way. To participate in discussions about topics related to the activity of a technical informatics engineer.

G6. SOLVENT USE OF THE INFORMATION RESOURCES: To manage the acquisition, structuring, analysis and visualization of data and information of the field of the informatics engineering, and value in a critical way the results of this management.

Full-or-part-time: 15h

Guided activities: 3h

Self study: 12h

GRADING SYSTEM

The grade is

$$N = 0.40 \cdot F + 0.25 \cdot P1 + 0.25 \cdot P2 + 0.10 \cdot D$$

where:

F = grade of the final exam

P1 = grade of the midterm exam (half course)

P2 = grade of the project (end of course)

D = note of the directed work

The midterm exam will be a computer exam that Haskell assesses. The final exam will be a written exam that will evaluate all the contents of the course. The practice is to use tools to generate compilers and Python to solve a case study.

The directed work consists in preparing a video and a written document about the properties of one or some programming languages. Assessment will be done by peer evaluation (co-evaluation). The qualifications of the transversal competences are obtained from the directed work.

BIBLIOGRAPHY

Basic:

- Wilhelm, R.; Maurer, D. Compiler design. Addison-Wesley, 1995. ISBN 978-0201422900.
- Aho, A.V.; Sethi, R.; Ullman, J.D. Compilers: principles, techniques, and tools. 2nd ed. Addison-Wesley, 2007. ISBN 9780321491695.
- Mitchell, J.C. Concepts in programming languages. Cambridge University Press, 2003. ISBN 978-0521780988.
- Scott, M.L. Programming language pragmatics. 4th ed. Morgan Kaufmann, 2016. ISBN 9780124104099.
- Thompson, S. Haskell: the craft of functional programming. 3rd ed. Addison-Wesley, 2011. ISBN 978-0201882957.
- Ruiz Jiménez, B.C. Razonando con Haskell: un curso sobre programación funcional. Thomson-Paraninfo, 2004. ISBN 9788497322775.