Course guide
270086 - CAP - Advanced Programming Concepts

Last modified: 13/07/2023

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 DELGADO PIN

Others: Primer quadrimestre:
JORDI DELGADO PIN - 11, 12

PRIOR SKILLS

Students should have enough knowledge of data structures and algorithms and object oriented programming and design.

REQUIREMENTS

- Prerequisite IES
- Prerequisite PROP

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:
CES1.1. To develop, maintain and evaluate complex and/or critical software systems and services.
CES1.3. To identify, evaluate and manage potential risks related to software building which could arise.
CES1.7. To control the quality and design tests in the software production

Generical:
G5. TEAMWORK: to be capable to work as a team member, being just one more member or performing management tasks, with the finality of contributing to develop projects in a pragmatic way and with responsibility sense; to assume compromises taking into account the available resources.

TEACHING METHODOLOGY

Teaching the course is structured in lectures and laboratory sessions.

Teachers will use lectures to introduce the essential contents of the course. In the laboratory sessions the contents of the course will be brought to the computer by carrying out practical problems. The laboratory classes will be a continuation of the lectures, where new concepts will be implemented as they appear in lectures.

LEARNING OBJECTIVES OF THE SUBJECT

1.Learn the basic concepts of prototype based programming, so that the student is able to understand how these concepts are implemented in different programming languages
2.Learn a language based on prototypes so that the student is able to get the general concepts in a concrete implementation (JavaScript)
3.Knowing what a closure is and some techniques associated with their use. Understanding what Continuations are and using them to implement several control structures
4.Learn a dynamic and object oriented programming language such as Smalltalk.
5.Learn the basics of computational reflection, so that the student is able to understand how these concepts are implemented in different programming languages
6.Knowing how computational reflection is implemented in Smalltalk: accessing and modifying members of classes at runtime, creating code at runtime, etc..
7.Being able to develop a computer program of small-middle size that uses some of the taught material

STUDY LOAD

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

Total learning time: 150 h

CONTENTS

Prototype based programming

Description:
Classes vs. prototypes. Inheritance vs. delegation

Control Structures: Continuations

Description:
Continuations are the result of being able to reify and eventually modify the execution stack of a program while it is running, but from the same language. We will explore the possibilities that this offers and the associated techniques.

Introduction to Smalltalk

Description:
Philosophy and origins. Encapsulation, members private / (protected) / public, inheritance, polymorphism, early / late binding. Patterns. Smalltalk, learning a Smalltalk environment

Reflection: General concepts

Description:
What is reflection?. Types of reflection: Introspection, intercession, reification. Implications of reflection for a programming language. Reflection in object-oriented programming.

Reflection in Smalltalk

Description:
Access to classes members at runtime. Patterns and Reflection. Creating reflective objects. Code Generation at runtime.

ACTIVITIES

Prototype based programming

Description:
The student should pay attention to the lecture and he/she should work through the exercises suggested by the lecturer.

Specific objectives:
1, 2

Full-or-part-time: 26h
Theory classes: 6h
Laboratory classes: 6h
Self study: 14h

Control Structures. Continuations

Description:
The student should pay attention to the lecture and he/she should work through the exercises suggested by the lecturer.

Specific objectives:
3

Full-or-part-time: 26h
Theory classes: 6h
Laboratory classes: 6h
Self study: 14h

Theory test

Specific objectives:
5, 6

Full-or-part-time: 16h
Guided activities: 2h
Self study: 14h

Practical exercise

Description:
Submission date: End of semester.

Specific objectives:
7

Related competencies :
G5. TEAMWORK: to be capable to work as a team member, being just one more member or performing management tasks, with the finality of contributing to develop projects in a pragmatic way and with responsibility sense; to assume compromises taking into account the available resources.

Introduction to Smalltalk

Description:
The student should pay attention to the lecture and he/she should work through the exercises suggested by the lecturer.

Specific objectives:
4

Full-or-part-time: 18h
Theory classes: 6h
Laboratory classes: 6h
Self study: 6h

Reflection: General concepts

Description:
The student should pay attention to the lecture and he/she should work through the exercises suggested by the lecturer.

Specific objectives:
5

Full-or-part-time: 14h
Theory classes: 2h
Laboratory classes: 2h
Self study: 10h

Reflection in Smalltalk

Description:
The student should pay attention to the lecture and he/she should work through the exercises suggested by the lecturer.

Specific objectives:
6, 7

Related competencies :
G5. TEAMWORK: to be capable to work as a team member, being just one more member or performing management tasks, with the finality of contributing to develop projects in a pragmatic way and with responsibility sense; to assume compromises taking into account the available resources.

Full-or-part-time: 36h
Theory classes: 8h
Laboratory classes: 8h
Self study: 20h

Final test

Specific objectives:
1, 2, 3

Full-or-part-time: 14h
Guided activities: 2h
Self study: 12h

GRADING SYSTEM

Grading the course will consist of two theoretical tests (T1 and T2), one mid-course and the other at the end, and one medium-sized practical work (Practice).

Then, the evaluation method would be:
0.8 * Theory + 0.2 * Practice
where:
Theory: 0.5 * T1 + 0.5 * T2

Teamwork:

Evaluated using a simple rubric that each group tutor group uses
to rank different aspects of teamwork of every member of the group.

BIBLIOGRAPHY

Basic:
- Black, A.P.; Ducasse, S.; Nierstrasz, O.; Pollet, D.; Cassou, D.; Denker, M. Pharo by example. Square Bracket Associates, 2009. ISBN 9783952334140.
- Noble, J.; Taivalsaari, A.; Moore, I. (Editors). Prototype-based programming: concepts, languages, and applications. Springer, 1999. ISBN 9814021253.

Complementary:
- Proceedings of OOPSLA '87, ACM SIGPLAN Notices. 22(12):147--155, 1987.

RESOURCES

Hyperlink:
- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.5354&rep=rep1&type=pdf- https://developer.mozilla.org/es/JavaScript