Course guide
820090 - PRE - Programming for Engineers

Last modified: 29/01/2026

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

Degree: BACHELOR'S DEGREE IN BIOMEDICAL ENGINEERING (Syllabus 2009). (Optional subject).
BACHELOR'S DEGREE IN CHEMICAL ENGINEERING (Syllabus 2009). (Optional subject).
BACHELOR'S DEGREE IN ELECTRICAL ENGINEERING (Syllabus 2009). (Optional subject).
BACHELOR'S DEGREE IN ENERGY ENGINEERING (Syllabus 2009). (Optional subject).
BACHELOR'S DEGREE IN INDUSTRIAL ELECTRONICS AND AUTOMATIC CONTROL ENGINEERING (Syllabus 2009). (Optional subject).
BACHELOR'S DEGREE IN MECHANICAL ENGINEERING (Syllabus 2009). (Optional subject).
BACHELOR'S DEGREE IN MATERIALS ENGINEERING (Syllabus 2010). (Optional subject).

Academic year: 2025 **ECTS Credits:** 6.0 **Languages:** Catalan, Spanish, English

LECTURER

Coordinating lecturer:

Others:

PRIOR SKILLS

This is a second programming course. Preferrably it is desirable that the students have acquired already the basic programming habilities (done in Informática Q1).

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:
1. Understand the basics behind the use and programming of PCs, operating systems, databases and software with applications in engineering.

Transversal:
2. EFFECTIVE USE OF INFORMATI0N RESOURCES - Level 3. Planning and using the information necessary for an academic assignment (a final thesis, for example) based on a critical appraisal of the information resources used.

TEACHING METHODOLOGY

There are weekly one theory session and one practice session in computer laboratory. In paralel the studends will propose and develop a project. Some practice sessions along the course will be devoted to the proposal. Collaborative learning. Project based learning (PBL).

LEARNING OBJECTIVES OF THE SUBJECT

Learning objectives:
1. Introducing the student to Object Oriented Programming.
2. Introducing the student to Event Oriented Programming and development of applications with graphical interface (GUI)

STUDY LOAD

Type	Hours	Percentage
Hours small group	30,0	20.00
Hours large group	30,0	20.00
Self study	90,0	60.00

Total learning time: 150 h

CONTENTS

Introduction

Description:
Tasks and evaluation method is explained.

Full-or-part-time: 10h
Theory classes: 2h
Laboratory classes: 4h
Self study : 4h

Analysis and life cicle

Description:
In this topic, students are provided with an overview of the software application development process.

Specific objectives:
-Describe the life cycle of a computer application and, by extension, of a software development project.
-Describe how to perform a requirements analysis.

Related activities:
Requirements analysis of the proposal.

Full-or-part-time: 6h
Theory classes: 2h
Self study : 4h

Application design

Description:
Diverse methodologies of application design are explained and practiced.

Specific objectives:
Object oriented design
Pattern design
Visual design

Related activities:
Object oriented design of the proposal
Pattern design of the proposal
Visual design of the proposal

Full-or-part-time: 42h
Theory classes: 16h
Laboratory classes: 2h
Self study : 24h

Programming in graphic environment

Description:
The necessary concepts to use visual programming elements are explained, and a project is developed.
Model View Presenter design patterns.

Full-or-part-time: 20h
Laboratory classes: 10h
Self study : 10h

Further Programming

Description:
Files
Exceptions and Error management
Files
Debugging
Version control with Git and Github

Full-or-part-time: 8h
Theory classes: 4h
Laboratory classes: 4h

Development

Description:
Development of the proposal by the student, culminating in a complete and executable program.

Full-or-part-time: 64h
Theory classes: 10h
Laboratory classes: 10h
Self study : 44h

GRADING SYSTEM

The subject is valued in successive deliverables of a project the student develops along the course. The various deliverables and their weights are the following:

Visual Design 15%
Object Oriented Design 20%
Patterns proposal 5%
Weekly Tasks 20%
Final Project 40%

EXAMINATION RULES.

There is no final exam.

BIBLIOGRAPHY

Basic:
- Yourdon, Edward. Techniques of program structure and design. Englewood Cliffs, NJ: Prentice-Hall, 1975. ISBN 013901702X.
- Parnas, D. L. "On the criteria to be used in decomposing systems in modules". Communicatios of the ACM [on line]. [Consultation: 22/04/2020]. Available on: https://doi-org.recursos.biblioteca.upc.edu/10.1145/361598.361623.- Summerfield, Mark. Rapid GUI programming with Python and Qt : the definitive guide to PyQt programming. Upper Saddle River, NJ: Prentice Hall, 2007. ISBN 9780132354189.
- Matthes, Eric. Python Crash Course : a hands-on, project-based introduction to programming [on line]. San Francisco: No Starch Press, 2016 [Consultation: 09/06/2020]. Available on: https://ebookcentral.proquest.com/lib/upcatalunya-ebooks/detail.action?docID=4503145. ISBN 9781593277390.