

Course guide

230901 - APR - Algorithms and Programming

Last modified: 24/05/2024

Unit in charge: Barcelona School of Telecommunications Engineering
Teaching unit: 701 - DAC - Department of Computer Architecture.

Degree: BACHELOR'S DEGREE IN ELECTRONIC ENGINEERING AND TELECOMMUNICATION (Syllabus 2018).
(Compulsory subject).

Academic year: 2024 **ECTS Credits:** 6.0 **Languages:** Catalan, Spanish

LECTURER

Coordinating lecturer: EVA RODRIGUEZ LUNA

Others: Primer quadrimestre:
JUAN CARLOS CRUELLAS IBARZ - 11
EVA RODRIGUEZ LUNA - 11, 12, 13

DEGREE COMPETENCES TO WHICH THE SUBJECT CONTRIBUTES

Specific:

CE2. (ENG) GREELC: coneixements bàsics sobre l'ús i programació dels ordinadors, sistemes operatius, bases de dades i programes informàtics amb aplicació a l'enginyeria. (Mòdul de formació bàsica).

Transversal:

CT4. (ENG) GREELEC: TREBALL EN EQUIP: ser capaç de treballar com a membre d'un equip interdisciplinar, ja sigui com un membre més o realitzant tasques de direcció, amb la finalitat de contribuir a desenvolupar projectes amb pragmatisme i sentit de la responsabilitat, assumint compromisos tenint en compte els recursos disponibles.

Basic:

CB2. (ENG) GREELEC: Que els estudiants sàpiguen aplicar els coneixements adquirits al seu treball o vocació d'una forma professional i tinguin les competències que solen demostrar-se per mitjà de l'elaboració i defensa d'arguments i la resolució de problemes dins de la seva àrea d'estudi.

CB3. (ENG) GREELEC: Que els estudiants tinguin la capacitat de reunir i interpretar dades rellevants (normalment dins de la seva àrea d'estudi) per emetre judicis que incloguin una reflexió sobre temes rellevants de caire social, científic o ètic.

TEACHING METHODOLOGY

MD1 Expositive method / Master class
MD2 Participative expositive class
MD3 Laboratory session
MD4 Cooperative work
MD5 Autonomous work
MD6 Problems/projects based learning

LEARNING OBJECTIVES OF THE SUBJECT

1. To be capable of using algorithmic techniques for solving mathematical problems of small and medium-sized complexity.
2. To develop (write, test and debug) computer programs that solve problems of small and medium-sized difficulty using the Python programming language.
3. Proper usage of the programming environment: edition, compilation, execution and debugging tools for the development and correction of programs.
4. Work in group. Definition of the tasks to be accomplished, equitable distribution of the tasks, and integration of the work carried out. Management of conflicts.

STUDY LOAD

Type	Hours	Percentage
Self study	85,0	56.67
Hours large group	39,0	26.00
Hours small group	26,0	17.33

Total learning time: 150 h

CONTENTS

Topic 1. Basic programming concepts

Description:

Basic programming concepts and terms: Algorithms, programs and programming languages. Stages in the development of a program. Structure of a program. Data types, constants, variables, expressions, assignments. Logical and relational operators. Operators' priority. Classes objects and methods.

Full-or-part-time: 22h 40m

Theory classes: 9h

Laboratory classes: 2h

Self study : 11h 40m

Topic 2. Basic algorithmic.

Description:

Basic mathematical and computational algorithms. Conditional sentences. Iterative sentences.

Full-or-part-time: 27h

Theory classes: 6h

Laboratory classes: 4h

Self study : 17h

Topic 3. Functions

Description:

Functions' calls.

Recursive design.

Specific Python modules.

Full-or-part-time: 30h 20m

Theory classes: 6h

Laboratory classes: 6h

Self study : 18h 20m



Topic 4. Structured data types

Description:

Representation of data structures using native Python types: Lists, Tuples, Dictionaries.
Data structures design.

Full-or-part-time: 28h 30m

Theory classes: 7h 30m

Laboratory classes: 4h

Self study : 17h

Topic 5. Algorithms of structured types

Description:

Basic algorithms: routing, search, insertion and extraction.
Advanced problems: matrixes, queues and stacks.

Full-or-part-time: 31h 30m

Theory classes: 7h 30m

Laboratory classes: 6h

Self study : 18h

Topic 6. Files and input/output.

Description:

Manipulation (creation/edition).
Traverse.

Full-or-part-time: 10h

Theory classes: 3h

Laboratory classes: 4h

Self study : 3h

GRADING SYSTEM

$\text{Final_mark} = \text{MAX}(0.6 * \text{Final_test}, 0.4 * \text{Final_test} + 0.2 * \text{Partial_test}) + 0.4 * \text{Lab_mark}$

$\text{Lab_mark} = 0.5 * \text{Lab_test} + 0.5 * \text{Project}$

BIBLIOGRAPHY

Basic:

- Lutz, M. Learning Python [on line]. 5th ed. Sebastopol, CA: O'Reilly, 2013 [Consultation: 25/09/2018]. Available on: <https://proquest.safaribooksonline.com/9781449355722?uicode=politicat>. ISBN 9781449355739.
- Downey, Allen; Elkner, Jeffrey; Meyers, Chris. How to think like a computer scientist : learning with Python. First edition. Wellesley, Massachusetts: Soho Books, 2002. ISBN 0971677506.