Last update: 03-07-2019

330509 - FI - Fundamentals of Computer Science

Coordinating unit: 330 - EPSEM - Manresa School of Engineering
Teaching unit: 750 - EMIT - Department of Mining, Industrial and ICT Engineering
Academic year: 2019
Degree: BACHELOR'S DEGREE IN AUTOMOTIVE ENGINEERING (Syllabus 2017). (Teaching unit Compulsory)
ECTS credits: 6 Teaching languages: Catalan

Teaching staff

Coordinator: Tarrés Puertas, Marta Isabel

Degree competences to which the subject contributes

Basic:

CB1. The students have demonstrated to possess and to understand knowledge in an area of study that starts from the base of the general secondary education, and is usually found to a level that, although it relies on advanced textbooks, also includes some aspects that involve knowledge from the vanguard of their field of study.

Specific:

CE3. Basic knowledge on the use and programming of computers, operating systems, databases and software with application in engineering.

Generical:

CG3. Knowledge in basic and technological subjects that will enable them to learn new methods and theories and give them the versatility to adapt to new situations.

Transversal:

1. EFFICIENT ORAL AND WRITTEN COMMUNICATION - Level 1. Planning oral communication, answering questions properly and writing straightforward texts that are spelt correctly and are grammatically coherent.
2. TEAMWORK - Level 1. Working in a team and making positive contributions once the aims and group and individual responsibilities have been defined. Reaching joint decisions on the strategy to be followed.
3. EFFECTIVE USE OF INFORMATION RESOURCES - Level 1. Identifying information needs. Using collections, premises and services that are available for designing and executing simple searches that are suited to the topic.
4. SELF-DIRECTED LEARNING - Level 1. Completing set tasks within established deadlines. Working with recommended information sources according to the guidelines set by lecturers.

Last update: 03-07-2019

330509 - FI - Fundamentals of Computer Science

Teaching methodology

In the two hours of theory sessions (EXP, RP), the professor alternates new theoretical concepts with examples and exercises. Lectures, in which the course topics are presented, explained and illustrated, are combined with student interaction regarding the various alternatives for solving practical cases. In the two hours of laboratory sessions (TP), some exercises are solved through collaboration. Students are expected to solve problems from a list during laboratory sessions and as part of their home study activities (TD). They are also advised to regularly consult their professor about the quality of their programs. Finally, students need to work in groups to solve a project (PR, PA). The final exam mark is awarded individually (EV).

MD1 Master class or lecture (EXP)
MD2 Problem solving and case study (RP)
MD3 Practical work in laboratory or workshop (TP)
MD4 Directed theoretical-practical work (TD)
MD5 Small-scale project, activity or assignment (PR)
MD6 Large-scale project or assignment (PA)
MD7 Assessment activities (EV)

Learning objectives of the subject

The course covers the basic concepts and techniques for building programs in imperative languages. By the end of the course, the student will be able to do the following:
1. Know all the constituents of programming languages: variables, types, expressions, conditional statements and iterations.

2. Use and design procedures and functions to find solutions to sub-problems.

3. Use the stepwise refinement method to solve programming problems of trivial and non-trivial complexity.

4. Know and be able to reproduce and use some fundamental algorithms for solving real problems.

5. Design, write, test, refine, document and maintain code in a high-level programming language to solve programming problems by applying algorithmic schemas and using data structures.

The programming language currently used is Python, although the emphasis is not on learning a specific language but on algorithmic problem solving and structured program construction.

Study load

Total learning time: 150h	Hours large group:	30h	20.00%
	Hours medium group:	0h	0.00%
	Hours small group:	30h	20.00%
	Guided activities:	0h	0.00%
	Self study:	90h	60.00%

Last update: 03-07-2019

330509 - FI - Fundamentals of Computer Science

Content

Topic 1: Introduction to programming

Learning time: 14h

Theory classes: 2h
Laboratory classes: 2h
Self study : 10h

Description:
This content is designed as a general introduction to the rapidly expanding field of computer science. Basic programming principles are introduced.
Glossary: computer, program, algorithm, bug, debugging, formal and natural programming languages, portability, interpreter, shell, script, syntax error, semantic error, runtime error, comment, value, variable, type, assignment, interactive mode, script mode, sentence, keyword, expression, operator, operand, rules of precedence, evaluate, composition, function, header, body function, block of code, function call, parameter, argument, return value, local variable.

Related activities:
All are detailed in “Planning of activities”.

Specific objectives:
1. To understand how to build a program.
2. To use the necessary tools: the console, editor and compiler.
3. To understand the syntax and semantics of basic expressions and instructions in a programming language.

Topic 2: Conditionals and iteration

Learning time: 26h

Theory classes: 8h
Laboratory classes: 8h
Self study : 10h

Description:
Control statements and iterative instructions.
Glossary: module, Boolean value, Boolean expression, Boolean operator, conditional expression, chained conditional, nested conditional, return statement, keyboard input, casting, None type, function composition, Boolean function, fruitful function, void function, iteration, infinite loop, local variable.

Related activities:
All are detailed in “Planning of activities”.

Specific objectives:
1. To understand the syntax and semantics of conditionals and iterations.
2. To build programs using conditionals and iterations.
3. To describe the main steps in the design of iterative algorithms.
4. To use functions and actions to develop programs; to understand the concepts of function, action and parameter passing.

330509 - FI - Fundamentals of Computer Science

Topic 3: Test-driven development

Learning time: 14h

Theory classes: 2h
Laboratory classes: 2h
Self study : 10h

Description:
Test-driven developments: unit test, doctest, nose and incremental development.
Glossary: unit test, doctest, nose test–driven development.

Related activities:
All are detailed in “Planning of activities”.

Specific objectives:
1. To implement a modular program in Python using test-drive development elegantly and in such a way that other programmers can understand what it does and modify it.
2. To understand and use doctests.
3. To write documentation that facilitates the use of a modular program written in Python by other programmers (docstrings and doctests used accordingly).

Topic 4: Strings

Learning time: 16h

Theory classes: 3h
Laboratory classes: 3h
Self study : 10h

Description:
Character data, string manipulation, problem solving with strings, slices, traversals, string modules.
Glossary: string, sequence, index, item, slice, traversal, IN operator, immutability, optional parameter, default value, string module, string formatting operator.

Related activities:
All are detailed in “Planning of activities”.

Specific objectives:
1. To design and implement algorithms using strings.
2. To use index and access operators.
3. To use the for iterator.
4. To understand the concept of string immutability.
5. To understand the main methods in string module.
6. To understand the use of the official Python documentation.
7. To understand predefined types in Python.

330509 - FI - Fundamentals of Computer Science

Topic 5: Lists

Learning time: 20h

Theory classes: 5h
Laboratory classes: 5h
Self study : 10h

Description:
Lists, ranges, clones, iterators, matrices, list comprehensions.
Glossary: list, nested list, list operation, list method, slice, range, mutability, list deletion, object, value, aliasing, cloning list, list iterator, list traversal, lists and functions, list parameter, nested list, matrix, lists and strings, list comprehension.

Related activities:
All are detailed in “Planning of activities”.

Specific objectives:
1. To know the data types typically used to represent and manage linear data structures.
2. To use list manipulation operators accordingly.
3. To use lists iterators.
4. To understand the concept of mutability in lists.
5. To design and implement programs using strings and lists.

Topic 6: Traversal and search schemes

Learning time: 14h

Theory classes: 2h
Laboratory classes: 2h
Self study : 10h

Description:
Sequences. Sequential traversal and search.
Glossary: sequence, traversal pattern, search pattern, for iterator, break sentence, else sentence in iterators.

Related activities:
All are detailed in “Planning of activities”.

Specific objectives:
1. To understand and use traversal schemes with while iterations.
2. To understand and use search schemes with while iterations.
3. To use search and traversal schemes with the for operator.

330509 - FI - Fundamentals of Computer Science

Topic 7: Modules and files

Learning time: 16h

Theory classes: 3h
Laboratory classes: 3h
Self study : 10h

Description:
File processing, module sys, argv
Glossary: module, import sentence, namespace, continue, object method, string method, list method, file, text file, open file, read/write file, close file, end of a file, directory, sys module, sys.argv

Related activities:
All are detailed in “Planning of activities”.

Specific objectives:
1. To implement functions in modules.
2. To use modules.
3. To build programs where the results are written in text files.
4. To build programs where the input data are stored in text files.
5. To build programs where the input data are provided by the command line.

Topic 8: Tuples

Learning time: 14h

Theory classes: 2h
Laboratory classes: 2h
Self study : 10h

Description:
Tuples and related operations.
Glossary: tuple, mutability and tuples, tuple assignment, tuples as return values

Related activities:
All are detailed in “Planning of activities”.

Specific objectives:
1. To understand the use of tuples.
2. To understand the concept of mutability in tuples.
3. To build list comprehensions.
4. To understand the concepts of the main sequence list, tuples and strings in Python and how they are related.

330509 - FI - Fundamentals of Computer Science

Topic 9: Dictionaries	Learning time: 16h Theory classes: 3h Laboratory classes: 3h Self study : 10h

Description:
Dictionaries and related operations, keys, values.
Glossary: dictionary, dictionary operation, dictionary method, aliasing and copying, key, key-value pair.

Related activities:
All are detailed in “Planning of activities”.

Specific objectives:
1. To build programs using dictionaries in Python.
2. To understand the concept of mutability in dictionaries.
3. To develop efficient solutions to problems using dictionaries when needed.

Last update: 03-07-2019

330509 - FI - Fundamentals of Computer Science

Planning of activities

Activity 1: Lectures

Hours: 12h
Theory classes: 12h

Description:
In the theory sessions, the professor will alternate new theoretical concepts with examples and exercises. Lectures, in which the course topics are presented, explained and illustrated, will be combined with student interaction regarding the alternatives for solving practical cases.

Support materials:
Book in web format.
Compulsory reading list.
List of problems.

Activity 2: Practical lectures

Hours: 12h
Theory classes: 12h

Description:
In the laboratory sessions and in home study, students are expected to solve problems. They are also advised to regularly consult their professor about the quality of their programs.

Support materials:
Book in web format.
Compulsory reading list.
List of problems.

Activity 3: Laboratory assignments

Hours: 41h
Self study: 15h
Laboratory classes: 26h

Description:
The laboratory sessions have two parts:
During the first hour, the professor describes practical issues regarding the programming environment, or some exercises are solved through collaboration, or some codes are analysed to identify errors, etc.
Then students devote the remaining two hours to solving problems, with the assistance of the professor if needed.

Support materials:
Book in web format
Compulsory reading list.
List of problems.
Free software previously installed on PCs.

Activity 4: Reading

Hours: 20h
Self study: 20h

330509 - FI - Fundamentals of Computer Science

Description:
The students study additional material provided in lectures.

Support materials:
Book in web format.
Compulsory reading list.
List of problems.

Activity 5: Written exercises	Hours: 25h Self study: 25h

Description:
The students solve homework problems, with the assistance of the professor if needed.

Support materials:
Book in web format.
Compulsory reading list.
List of problems.

Activity 6: Project	Hours: 28h Laboratory classes: 4h Theory classes: 4h Self study: 20h

Description:
The programming project integrates knowledge and skills of the entire course. In a previous laboratory session, teams of students will be trained to solve the project.

Activity 7: Exam	Hours: 12h Theory classes: 2h Self study: 10h

Description:
Final exam mark.

Last update: 03-07-2019

330509 - FI - Fundamentals of Computer Science

Qualification system

In this course three types of assessment are used:

1. Continuous assessment: A
2. Project assessment: P
3. Final examination: F

Final marks for the course will be determined using the following weighting:

Final = 0.35A + 0.25P + 0.40F

Bibliography

Basic:

Downey, Allen. Python for software design: how to think like a computer scientist. Cambridge: Cambridge University, 2009. ISBN 9780521725965.

Complementary:

Pilgrim, Mark. Dive into Python. New York: Apress, 2004. ISBN 1590593561.

Guzdial, Mark; Ericson, Barbara. Introduction to computing & programming in Python: a multimedia approach. 2nd ed. Upper Saddle River: Pearson/Prentice Hall, 2010. ISBN 9780136060239.

Others resources: