Universidad de Alcalá

TEACHING GUIDE

Principles of Python
(Basic programming)

Transversal Subject

Universidad de Alcalá

Academic Year 2021/2022

1st Semester

Approved by the School Board on July 6, 2021

Universidad de Alcalá

TEACHING GUIDE

Course Name:	**Principles of Python (basic programming)**
Code:	**100263**
Degree in:	
Department and area:	**Signal Theory and Communications/ Signal Theory and Communications**
Type:	**Transversal**
ECTS Credits:	**6**
Year and semester:	**1st semester**
Teachers:	**Carolina Gil Marcelino**
Tutoring schedule:	It will be communicated at the beginning of the course
Language:	**English**

1. PRESENTATION

The essential objective of this course is that the student learns and identifies the relevant information of a problem and its representation and manipulation in programming. The course aims to develop the following fundamental computer programming skills:

- Understanding the syntactic and semantic aspects of Python programming language.

- Articulation of commands, data structures and basic programming structures to build solutions to simple problems.

- Modeling and implementation of modularized code for non-elementary problems.

- Construction of organized, reusable and readable code, following the principles of good programming practices.

In summary, the student will acquire a general and initial knowledge about structured imperative programming using Python language.

Learning how to program can open the door to a better job, new areas of work or even a new way of seeing how it works all around us. The main objective of the course is to develop the skills for building readable and modular programs in Python. The modules will be implemented by Python functions. The course starts with teaching Python functions. The introduction of this content as a basis for teaching programming is justified by the concept of function being the most concrete, within the universe of students, of something that relates information (data) with sequences

of operations that transform the information (s) of entry into the desired (outgoing) information is the function. In general, students have already seen mathematical functions along the student path and are able to bring some intuitions from mathematics to the functions of programming.

After understanding the concept of function and its use, we move on to teaching decision structures (conditional). Considerable time is spent working on building tests and using the decision structure. The Lista type is presented after conditional structure. The repetition structures are introduced in the second half of the course, when it is expected that the student is already mature in the understanding that the programming language is a formal, unambiguous, and deterministic language. At this point, the student must be able to critically analyze and debug his own code. To develop this competence, in addition to practical exercises and program building, several "test table" exercises are presented throughout the course. Manipulation of lists and matrices (list of lists) are the main motivators for the construction of functions that make use of repetition structures, reinforcing the importance of this content having been seen immediately before. And after working with manipulating lists and repetition structures, we move on to another type of compound data: dictionaries. This type allows us to work on the concept of indexing.

The last content covered in the course is what we call "user interaction", being the construction of a main program and the use of input and output commands. Thus,
the main objective of the assignment is that the student learns and identifies the relevant information of a problem and its respective representation and manipulation in the programming.

3. SKILLS

Basic, general and transversal competences.

This subject contributes to acquiring the following basic, general and transversal competences defined in section 3 of the Annex to Order CIN / 352/2009:

TRU1 - Analysis and synthesis capacity.
TRU2 - Oral and written communication.
TRU3 - Information management capacity.
TRU4 - Self-learning.
TRU5 - Teamwork.

Learning Outcomes

Upon successful completion of this course, students will be able to:

RA1. Understand and build modular programs capable of assisting in calculations, data analysis, and the like at a basic level.
RA2. Identify and apply concepts and techniques from other subjects to build computational solutions for mathematical problems.

4. CONTENTS

Content Blocks	Total number of hours
Topic 1: Fundamentals of Python • Introduction: algorithm concepts, examples, programming languages and paradigms. Python language and versions. Laboratory practice.	4 hours
Topic 2: Python Functions. • Function definition, call functions, default arguments, types, modules: math. Examples. Laboratory practice.	4 hours
Topic 3: Python - Data types and Conditional structure: • Numerical (int, float, complex), String (str), Boolean, Boolean expressions, conditional struct (if, else, elif), examples. Laboratory practice.	4 hours
Topic 4: Python - Variables and Strings: • Variables (attribution), Variables (names), Variables (scope), Strings (index), Strings (slice), examples. Laboratory practice.	4 hours
Topic 5: Python - String manipulation, Tuple, List • String manipulation (lower, upper, count, index, format), Tuple (types, len, concat), examples. Laboratory practice.	4 hours
Topic 6: Python - List • List (types, concat, range), malipulation, slice, examples. Laboratory practice.	4 hours
Topic 7: Python - Loop (while): • String manipulation (lower, upper, count, index, format), Tuple (types, len, concat), List (types, concat, range), examples. Laboratory practice.	4 hours

Topic 8: Python - Loop (for): • String manipulation (lower, upper, count, index, format), Tuple (types, len, concat), List (types, concat, range), examples. Laboratory practice.	4 hours
Topic 9: Python - Nested loops and Matrices: • Nested loops with while, matrices with for, examples. Laboratory practice.	4 hours
Topic 10: Python - Data struct - Dictionary: • Concept, using list, manipulations, examples, Laboratory practice.	4 hours
Topic 11: Python - User Interaction (Input and Output): • Concepts, print, "\n", str.format, main() function, input, examples . Laboratory practice.	4 hours
Topic 12: Python - User Interaction (Input and Output): • concepts, why use? how to make? main function (), examples. Laboratory practice.	4 hours
Final work: Implementation of a practical case	8 hours

5. TEACHING - LEARNING METHODOLOGIES. FORMATIVE ACTIVITIES.

5.1. Credits Distribution

Number of on-site hours:	42 h + 3h (exams)
Number of hours of student work:	105 h
Total hours	150 h

5.2. Methodological strategies, teaching materials and resources

In the teaching and learning process the following training activities will be undertaken:

- Theoretical Classes and example solving.
- Practical Classes: laboratory and exercise solving.
- Carrying out practical cases.

Throughout the course the student will be proposed activities and tasks, so that he can consolidate the concepts acquired.

Throughout the learning process in the subject, the student must make use of different bibliographic or electronic sources and resources.

The teaching staff of this subject will provide their own materials elaborated specifically for it so that the student can meet the objectives of the subject and achieve the planned competencies.

6. ASSESSMENT: procedures, evaluation and grading criteria

6.1. Evaluation criteria

The objective of the evaluation process is to analyze what skills the student has acquired and to what degree. The tests and procedures detailed below are proposed in order to extract and assess the evaluation criteria that are set out below:

CE1. The student knows how to construct functions and use conditional control.
CE2. The student knows how to use strings, tuples and lists.
CE3. The student knows how to use repeating structures, nested loops and matrices.
CE4. The student knows how to use dictionaries, modularization and user interaction.

6.2. Evaluation instruments

In the evaluation of the subject the following evaluation instruments will be used:

1. Intermediate Assessment Tests (IAT). They will be individual and in writing. They will not release matter. They will consist of solving problems and questions to assess the extent to which the student is progressing in meaningful learning.

2. Final Test (EF). It will be a final exam and will be done in writing and individually. Its objective is to evaluate that the student has acquired an integrated knowledge of the subject as a whole. It will consist of solving problems and questions that involve all the topics of the complete subject.

3. Subject work (SW): Resolution of a practical case in which specific aspects of the syllabus will have to be applied.

“A student who follows the continuous evaluation procedure will be considered not to have taken the ordinary call if he or she does not take exam IAT1."

6.3. GRADING CRITERIA

5.3.1. Ordinary call:
a) According to the continuous evaluation model.

Skill	Learning Outcomes	Evaluation criteria	Grading Tool	Weight in the final grade
TR1, TR2, TR3	RA1, RA2	CE1, CE2	IAT1	20%
		CE3, CE4	IAT2	20%
TR1, TR2, TR3	RA1, RA2	CE1, CE2	EF	40%
		CE3, CE4		
TR4, TR5	RA1, RA2	CE1, CE2	SW	20%
		CE3, CE4		

"A student who follows the continuous evaluation procedure will be considered not to have taken the ordinary call if he or she does not take exam IAT1."

b) <u>According to the final evaluation model.</u>

Skill	Learning Outcomes	Evaluation criteria	Grading Tool	Weight in the final grade
TR1, TR2, TR3	RA1, RA2	CE1, CE2	EF	100%
		CE3, CE4		

"A student who follows the continuous evaluation procedure will be considered not to have taken the ordinary call if he or she does not take exam IAT1."

5.3.2. Extraordinary call:

Finally, students who do not pass the subject in the ordinary call will have to pass it in the extraordinary call, which will obey the same scheme as the final evaluation model.

7. BIBLIOGRAPHY

Basic Bibliography

- Delgado, C. A. D. M. ; Silva, J. C. P. ; Mascarenhas, F. ; Duboc, A. L.C. L. The teaching of functions as the first step to learn imperative programming. Available in en: https://ebooks.pucrs.br/edipucrs/anais/csbc/assets/2016/wei/41.pdf
- Source material from the Institute of Computing, UFRJ, Brazil, Available in: https://dcc.ufrj.br/~pythonufrj/python1_37.html
- Severance, C.R. "Python for Everybody: Exploring Data Using Python 3". Editorial Support: Elliott Hauser, Sue Blumenberg. Cover Design: Aimee Andrion.

Additional Bibliography

- Beazley, D. and Jones, B. "Python Cookbook". O'REILY. 2013.

Disclosure Note

The University of Alcalá guarantees to its students that, if due to health requirements the competent authorities do not allow the total or partial attendance of the teaching activities, the teaching plans will achieve their objectives through a teaching-learning and evaluation methodology in online format, which will return to the face-to-face mode as soon as these impediments cease.