SUBJECT TEACHING GUIDE

G47 - PROGRAMMING

Double Degree in Physics and Mathematics
Degree in Mathematics
Degree in Mathematics
Academic year 2023-2024

1. IDENTIFYING DATA				
Degree	Double Degree in Physics and Mathematics Degree in Mathematics	Type and Year	Core. Year 1 Core. Year 1	
Faculty	Faculty of Sciences			
Discipline	Basic Related Subject Area: Basic Module			
Course unit title and code	G47 - PROGRAMMING			
Number of ECTS credits allocated	6	Term	Semester based (2)	
Web	https://www.istr.unican.es/asignaturas/prog_fis_mat/			
Language of instruction	Spanish	English Friendly	Yes	Mode of delivery	Face-to-face

Department	DPTO. INGENIERÍA INFORMÁTICA Y ELECTRÓNICA
Name of lecturer	MICHAEL GONZALEZ HARBOUR
E-mail	michael.gonzalez@unican.es
Office	Facultad de Ciencias. Planta: + 3. DESPACHO PROFESORES (3055)
Other lecturers	JOSE JAVIER GUTIERREZ GARCIA JOSE IGNACIO ESPESO MARTINEZ JOSE CARLOS PALENCIA GUTIERREZ ADOLFO GARANDAL MARTIN

3.1 LEARNING OUTCOMES
- • To be able to design, implement and test algorithms and simple programs using a programming language. • To know how to code classes and objects by applying object-oriented programming. • To know basic algorithms applicable to structured data (such as iteration, searches, matrix operations) • To use a programming environment to edit, compile and run programs. • To know how to use operating systems to perform basic tasks from the command line .

4. OBJECTIVES

- To know and understand the syntax and semantics of the statements of an imperative programming language .
- To know the main algorithmic constructs : sequence, conditionals, iteration and recursion
- To understand the concepts of class and object as constituent elements of the programs
- To know and use basic data types, tables and matrices, and to learn basic algorithms
for their manipulation (iteration, searches, simple sorting).
- To understand the concepts of method and parameter passing .
- To know the principles of modularity and abstraction to create simple program modules
- To know how to perform error handling with exceptions
- To acquire basic knowledge of object-oriented programming
- To know the principles of input / output : interactive and with files
- To use basic operating system commands

- To design small algorithms using pseudocode .
- To be able to code and test algorithms using an imperative programming language .
- To use a development system to edit , compile and run programs.
- To use an operating system at the user level .
- To create program modules, separating the phases of design and implementation.
- To code programs using an object-oriented language.
- To implement simple programs that are reliable and easy to understand .
- To use predefined program modules .
- To apply simple test strategies for a program module .
- To know how to document a programming project .

6. COURSE ORGANIZATION	
CONTENTS	
1	THEMATIC UNIT 1: Programming in Python
1.1	0. Introduction to the course
1.2	1. Introduction to programming languages. Programming languages. Compilers and interpreters. The software life cycle. The Python Language. Concept of algorithm. Encapsulating data and algorithms. Program structure. Functions. Coding style.
1.3	2. Data and expressions. Numbers. Operators and expressions. Variables. Booleans. Strings. Using mathematical functions. Variables and parameter passing. Lists and tuples.
1.4	3. Classes. Classes and objects.Defining classes. Creating and using objects. Instance and class attributes and methods. Name spaces. Modules and packages.
1.5	4. Algorithmic structures. Conditional instruction. Multiway branch statement. Loop statements. Recursion. Describing algorithms with pseudocode.
1.6	5. Data structures. Tables. Iteration algorithms. Search algorithms. Sets. Multidimensional tables. The numpy package. Dictionaries. Enumerate types
1.7	6. Error handling. Exceptions. Exception handling. Exception handling patterns. Throwing exceptions. Using our own exceptions. Cleaning actions.
1.8	7. Input / Output. Formatted text output. Formatted text input. Files. Reading text files. Writing text files.
1.9	8. Inheritance and polymorphism. Class hierarchies and inheritance. Abstract classes. Polymorphism.
2	THEMATIC UNIT 2: Tools
2.1	9. Using operating systems. Common operating systems.The file system. The shell. Running programs. The graphical file manager. Scripts. Using a USB memory.
2.2	10. Using an integrated software development environment. Integrated development environment. Managing projects. Analyzing, loading and executing programs. Debugging. Document generation.

7. ASSESSMENT METHODS AND CRITERIA				
Description	Type	Final Eval.	Reassessn	%
Presentation of solutions to short exercises solved at home	Others	No	Yes	10,00
Final written exam. Notes and books are allowed during the final exam, but not electronic media such as laptops, tablets and mobile phones.	Written exam	Yes	Yes	50,00
Theory: classroom participation	Others	No	No	5,00
Assessment of lab assignments and reports. There will be penalties for late reports.	Laboratory evaluation	No	Yes	35,00
TOTAL				100,00

Observations

The grade for the course consists of three parts:

a) classroom participation: 5%
b) continuous assessment of exercises and lab assignments: 45%
This part of the assessment consists of two parts:
b.1) Short exercises (10%)
b.2) Lab assignments (35%)
c) Final exam: 50%

Part b, continuous assessment of exercises and lab assignments, can be replaced by a lab exam in the extraordinary evaluation period.

To pass the subject is necessary to score a minimum mark of 4 in both part b) and part c). If only one of these parts is passed in the ordinary evaluation period, the mark obtained for that part will be saved for the extraordinary evaluation period.
Lab assignments, exercises and exams are individual. It is considered cheating:
- to solve exercises or lab assignments in a group
- to exchange exercises between classmates
- that other people or an AI do the exercises or most of them

Observations for part-time students

For part-time students not being present at the exercises classes will have no penalty.
For part-time students continuous assessment of short exercises and lab assignments can be replaced by a lab exam, both in the ordinary and extraordinary evaluation periods: For these students the weight of this exam is 45%, and the weight of the final exam is 55%. There will be no evaluation of the classroom participation.

8. BIBLIOGRAPHY AND TEACHING MATERIALS
BASIC
Title: Python in a Nutshell: A Desktop Quick Reference 3rd Edition by: Alex Martelli (Author), Anna Ravenscroft (Author), Steve Holden Publisher: O'Reilly Media; 3 edition (May 4, 2017) ISBN-10: 144939292X ISBN-13: 978-1449392925
Tutorial de python 3: https://docs.python.org/3/tutorial/ http://docs.python.org.ar/tutorial/3/index.html
Title: Introducción a la programación con Python 3 By: Andrés Marzal Varó, Isabel Gracia Luengo, Pedro García Sevilla Editor: Universitat Jaume I, 2014 ISBN: 978-84-697-1178-1 http://repositori.uji.es/xmlui/bitstream/handle/10234/102653/s93.pdf