COMPUTER PROGRAMMING 1

Bachelor in Applied Mathematics BAM SEP-2023 CP1-AM.1.S.A

Area Others

Number of sessions: 30

Academic year: 23-24

Degree course: FIRST

Number of credits: 6.0

Semester: 1º

Category: BASIC

Language: English

Professor: **DANIEL PRECIOSO GARCELÁN**

E-mail: dprecioso@faculty.ie.edu

Daniel Precioso

Daniel Precioso received his degree in Physics from the Universidad Complutense de Madrid, Spain, and his Master's degree in Statistical and Computational Information Processing from the Universidad Politécnica de Madrid, Spain. He moved to Cádiz to get his PhD in Machine Learning and Data Science from Universidad de Cádiz publishing in reputed journals in the field and spending time abroad visiting renowned universities in Canada. During his Phd years he also participated in some international Data Science competitions, in collaboration with other researchers. His recent research efforts focus mainly on applying Data Science to the Blue Economy, mainly via the optimization of maritime shipping routes.

Education:

- Ph.D in Machine Learning and Data Science, Universidad de Cádiz.
- MSc in Statistical and Computational Information Processing, Universidad de Politécnica de Madrid.
- Bs in Physics, Universidad Complutense de Madrid.

dprecioso@faculty.ie.edu

SUBJECT DESCRIPTION

Computer programming is the art and science of writing software programs that instruct computers to perform various tasks. It is a skill that is essential in today's world and is used in almost every aspect of our lives, from mobile applications to scientific research. In the field of Mathematics, programming is a powerful tool for modeling, simulating, and solving complex problems.

This course is designed as an introduction to programming for students who have never programmed before, with a focus on applications in Mathematics. Students will learn the fundamental concepts and skills necessary to develop software programs, including the Python programming language, control structures, data types, and algorithms. They will also learn to develop efficient and scalable programs by employing best practices in programming.

Throughout this course, students will be provided with hands-on programming exercises that apply programming concepts to real-world problems in Mathematics and other related subjects. They will learn to design and implement algorithms to solve problems and to write efficient and readable code using common software engineering practices.

The course covers a comprehensive introduction to the basic concepts of programming, including variables, data types, and control structures (if/else statements, loops). Additionally, students will learn fundamental algorithms and data structures, such as arrays, lists, tuples, and dictionaries, and modular programming and functions, including parameter passing. The subject also emphasizes debugging techniques to help students identify and solve software bugs.

By the end of the course, students will have a strong foundation in programming and software development, with a focus on its applications in Mathematics. They will be able to write programs that can automate tasks, process data, and perform complex computations. They will also have a fundamental understanding of programming languages and the tools and techniques used to develop software applications, which will be invaluable in their future studies and careers in Mathematics, Data Science, and beyond.

LEARNING OBJECTIVES

The main objective of this course is to provide students with a strong foundation in computer programming that can be applied to solve problems in various fields, including Mathematics, Statistics, Engineering, and Science.

At the end of the course, students should be able to:

- Write simple programs using the Python programming language to solve mathematical problems.
- Understand the basic concepts of programming such as variables, data types, control structures, and functions.
- Develop algorithms and implement data structures such as arrays, lists, tuples, and dictionaries to solve complex problems.
- Apply best practices in programming to write efficient, scalable, and maintainable code.
- Debug programs using common techniques to identify and fix errors in code.
- Analyze real-world problems and design programs to automate tasks and perform computations.
- Gain exposure to other programming tools used in software development.

Additionally, the course will focus on the acquisition or reinforcement of generic skills:

- The ability to summarize and present information in a clear and concise manner.
- The ability to identify patterns and abstract models to solve real-world problems.
- The ability to work collaboratively in teams to develop software applications.
- The ability to adapt to new technologies as they emerge.

TEACHING METHODOLOGY

All sessions for this course will be live-in sessions taken at the IE Segovia's campus. There will be thirty 90 minutes classes distributed in the following way:

- 28 Teaching sessions
- 3 Exam review sessions
- 1 Midterm exam
- 1 Final exam

Each lecture will be based on a combination of theoretical explanations and several practical exercises. Each computing concept will be followed immediately by one or more examples related to the fields of interest of the students. Emphasis will be placed on applying the computing knowledge gained from this course to tackle problems related to other subjects within the same Bachelor degree. Student participation is considered very important to acquire the skills needed to solve exercises. Problem sets and brief quizzes will be used along the course.

Problem sets are in-depth problems that will be uploaded to campus online. I strongly recommend that you do the exercises given as homework during the course and not leave them for a date close to the exam. Though you are strongly encouraged to work with others on understanding the lecture material and attempting the regular assignments, the intention is that you work alone on the Problem Sets, which are designed to give you feedback on how you are progressing.

Brief quizzes will be given throughout the semester. These quizzes are announced and will cover previously taught material. These quizzes are meant to test your overall understanding of the material and will help the professor assess the overall performance and evolution of the class.

Bringing your **laptop** is mandatory to all sessions, as it will be used to code in Python. A Google account and access to **Google Drive** is also mandatory, to run the Collab notebooks that will be used regularly during sessions. In addition, **VSCode** will need to be installed in the student's laptops, however the installation will be covered as part of the course. Configuration of Python IDEs in the student's laptops needs to be done selecting English as the working language to facilitate the collaboration of students and faculty. Some activities will require the installation of secure open-source libraries for Python into the student's laptops.

IE University teaching method is defined by its collaborative, active, and applied nature. Students actively participate in the whole process to build their knowledge and sharpen their skills. Professor’s main role is to lead and guide students to achieve the learning objectives of the course. This is done by engaging in a diverse range of teaching techniques and different types of learning activities such as the following:

Learning Activity	**Weighting**	**Estimated time a student should dedicate to prepare for and participate in**
Lectures	20.0 %	30.0 hours
Discussions	13.33 %	20.0 hours
Exercises in class, Asynchronous sessions, Field Work	30.0 %	45.0 hours
Group work	10.0 %	15.0 hours
Individual studying	26.67 %	40.0 hours
TOTAL	100.0 %	150.0 hours

INTRODUCTION TO THE PROGRAM

Disclaimer: *The following description of the material covered is tentative. An attempt will be made to cover all listed topics and to include other advanced topics that will help the student throughout his/her career in mathematics. However, the pace of the classes will depend on group performance, which may introduce some variations in the syllabus.*

This course will serve as an introduction to computer programming on Python. In this context, the course is divided into five modules, each module consisting of 3 to 6 live in-person sessions, combining concepts and practice. The rest of the sessions are dedicated to two exams and three exam review sessions.

- Module 1: FUNDAMENTALS
- Module 2: FUNCTIONS
- Module 3: BASIC DATA STRUCTURES
- Module 4: PYTHON ECOSYSTEM
- Module 5: VECTOR PROGRAMMING

When needed, extra materials will be published in the Campus Documentation Section to cover specific topics.

PROGRAM

SESSION 1 (LIVE IN-PERSON)

Module 1 / 5: Fundamentals

Topics treated:

- Introduction to the subject.
- Introduction to Python and Google Collab notebooks.
- Numeric variables in Python: integers and floats.
- Operations with numeric variables (arithmetic operators).

Pre and post-work:

- No pre-work needed.
- After the session problem sets and instructions about how to work with them will be available online.

SESSION 2 (LIVE IN-PERSON)

Module 1 / 5: Fundamentals

Topics treated:

- Text variables in Python: strings.
- Operations with strings: indexing, slicing and concatenation.
- String methods: length(), upper(), lower(), strip(), split().

Pre and Post-work:

- Work on selected exercises.

SESSION 3 (LIVE IN-PERSON)

Module 1 / 5: Fundamentals

Topics treated:

- Flow control: the if/else statement.
- Logical operators: and, or, not.
- Debugging in Python.

Pre and post-work:

- Work on selected exercises.

SESSION 4 (LIVE IN-PERSON)

Module 1 / 5: Fundamentals

Topics treated:

- Looping in Python: for and while loops. Existing a loop using break and continue.
- Combining loops and flow control in Python.

Pre and Post-work:

- Work on selected exercises.

SESSION 5 (LIVE IN-PERSON)

Module 2 / 5: Functions

Topics treated:

- What is a function?
- Basic build-in functions in Python: print(), len(), type()…

Pre and Post-work:

- After the session a new problem set will be available.

SESSION 6 (LIVE IN-PERSON)

Module 2 / 5: Functions

Topics treated:

- Creating user-defined functions with “def”.
- Positional arguments and their order, return values.
- Keyword arguments and their default values.

Pre and Post-work:

- Work on selected exercises.

SESSION 7 (LIVE IN-PERSON)

Module 2 / 5: Functions

Topics treated:

- What is a docstring? What are type hints?
- The Python scope. Understanding the global and local namespaces.
- Accessing variables in different scopes.

Pre and Post-work:

- Work on selected exercises.

SESSION 8 (LIVE IN-PERSON)

Module 2 / 5: Functions

Topics treated:

- Generator functions: the yield statement.
- Examples and applications of the concepts covered thus far, with a focus on their potential applications to other subjects in Applied Mathematics.

Pre and Post-work:

- Work on selected exercises.

SESSION 9 (LIVE IN-PERSON)

Module 3 / 5: Basic data structures

Topics treated:

- Introduction to all data structures and properties of each: lists, tuples, sets and dictionaries.
- Lists in Python: how to create, index and update.
- Build-in lists methods, and functions relevant to them.
- Iterating over a list using Python loops.

Pre and Post-work:

- After the session a new problem set will be available.

SESSION 10 (LIVE IN-PERSON)

Module 3 / 5: Basic data structures

Topics treated:

- Tuples in Python.
- Difference between lists and tuples. How to convert between one another.
- What is list comprehension?

Pre and Post-work:

- Work on selected exercises.

SESSION 11 (LIVE IN-PERSON)

Module 3 / 5: Basic data structures

Topics treated:

- Sets in Python.
- Difference between lists, tuples and sets. How to convert between one another.
- Operating with sets: union, intersection, subtraction and symmetric difference.

Pre and Post-work:

- Work on selected exercises.

SESSION 12 (LIVE IN-PERSON)

Module 3 / 5: Basic data structures

Topics treated:

- Dictionaries in Python. What are the key-value pairs?
- Working with dictionaries: how to update them and pop keys.
- How to create dictionaries from lists, tuples and sets.

Pre and Post-work:

- Work on selected exercises.

SESSION 13 (LIVE IN-PERSON)

Module 3 / 5: Basic data structures

Topics treated:

- How functions may affect data structures: the effects of mutability.
- Examples and applications of data structures, with a focus on their potential applications to other subjects in Applied Mathematics.

Pre and Post-work:

- Work on selected exercises.

SESSION 14 (LIVE IN-PERSON)

Review for the midterm exam.

Scope: Modules 1, 2 and 3.

SESSION 15 (LIVE IN-PERSON)

Mid-term exam.

Scope: Modules 1, 2 and 3.

SESSION 16 (LIVE IN-PERSON)

Module 4 / 5: The Python ecosystem

Topics treated:

- Introduction to Python IDEs: PyCharm, Spyder, VSCode.
- Downloading and installing VSCode.
- How to use Jupyter notebooks in VSCode.

Pre and Post-work:

- After the session a new problem set will be available.

SESSION 17 (LIVE IN-PERSON)

Module 4 / 5: The Python ecosystem

Topics treated:

- Writing scripts in Python.
- Running scripts: execution order and the “main” function.

- Code reusability: how to import functions from other scripts.

Pre and Post-work:

- Work on selected exercises.

SESSION 18 (LIVE IN-PERSON)

Module 4 / 5: The Python ecosystem

Topics treated:

- How to read text files and json files, using the open() function.
- How to store data structures (list, dictionaries) into files.
- What is a library? Example of a standard library in Python: the “random” library.
- Using the “random” library to simulate random events and running Monte Carlo experiments.

Pre and Post-work:

- Work on selected exercises.

SESSION 19 (LIVE IN-PERSON)

Module 4 / 5: The Python ecosystem

Topics treated:

- Introduction to Python environments.
- Managing environments with Conda.
- How to Install Python libraries with Pip and Conda.

Pre and Post-work:

- Work on selected exercises.

SESSION 20 (LIVE IN-PERSON)

Module 5 / 5: Vector programming

Topics treated:

- Introduction to NumPy.
- Creating and manipulating arrays in NumPy.
- Array indexing and slicing.

Pre and Post-work:

- After the session a new problem set will be available.

SESSION 21 (LIVE IN-PERSON)

Module 5 / 5: Vector programming

Topics treated:

- Arithmetic operations on NumPy.
- Broadcasting in NumPy.
- NumPy build-in methods and functions.

Pre and Post-work:

- Work on selected exercises.

SESSION 22 (LIVE IN-PERSON)

Module 5 / 5: Vector programming

Topics treated:

- Reshaping and resizing arrays.
- Joining and splitting arrays.
- Changing array shapers and dimensions.

Pre and Post-work:

- Work on selected exercises.

SESSION 23 (LIVE IN-PERSON)

Module 5 / 5: Vector programming

Topics treated:

- Loading and saving arrays to file.
- Advanced indexing techniques (e.g., boolean indexing, fancy indexing).

Pre and Post-work:

- Work on selected exercises.

SESSION 24 (LIVE IN-PERSON)

Module 5 / 5: Vector programming

Topics treated:

- Matrix multiplication and dot products.
- Solving linear systems of equations.
- Eigenvectors and eigenvalues.

Pre and Post-work:

- Work on selected exercises.

SESSION 25 (LIVE IN-PERSON)

Module 5 / 5: Vector programming

Topics treated:

- Descriptive statistics: mean, median, standard deviation.
- Correlation and covariance.
- Hypothesis testing with NumPy.

Pre and Post-work:

- Work on selected exercises.

SESSION 26 (LIVE IN-PERSON)

Module 5 / 5: Vector programming

Topics treated:

- Generating random numbers with NumPy.
- Monte Carlo simulations with NumPy.
- Simulating statistical distributions with NumPy.

Pre and Post-work:

- Work on selected exercises.

SESSION 27 (LIVE IN-PERSON)

Module 5 / 5: Vector programming

Topics treated:

- Review and summary of NumPy.
- The importance of vectorization.
- Examples and applications of arrays, with a focus on their potential applications to other subjects in Applied Mathematics.

Pre and Post-work:

- Work on selected exercises.

SESSION 28 (LIVE IN-PERSON)

Review for the final exam.

Scope: Modules 1 to 5.

SESSION 29 (LIVE IN-PERSON)

Review for the final exam.

Scope: Modules 1 to 5.

SESSION 30 (LIVE IN-PERSON)

Final exam

Scope: Modules 1 to 5.

EVALUATION CRITERIA

CLASS PARTICIPATION

Class participation will be worth 10% of the overall grade. This entails being present in class and on time. Students are expected to participate actively during lectures with questions and remarks. There may be a penalty if you create a disruption or talk excessively during class.

HOMEWORK

Each module has its own problem set including several exercises. The student is expected to work on them and to turn in selected exercises that will be announced throughout the curse. Most exercises are individual work but some might involve group work. Each problem set will be graded after the deadline and your overall "Homework" grade is worth 30% of the final grade. The “Homework” grade will be the arithmetic mean of all the problem sets' marks.

QUIZZES + MIDTERM EXAM

It is worth 30% of the final grade. We will have several small quizzes scattered throughout the course. They will be based on exercises similar to the ones we will see in class.

FINAL-EXAM

It is worth 30% of the overall grade. You need to score at least 3.5 on the final exam to pass the overall course, even if you have already passed the course through the other course assessments. Information about the detailed characteristics of the final-exam will be given at the beginning of the semester.

criteria	percentage	Learning Objectives	Comments
Final Exam	30 %		
Quizzes + Mid-term exam	30 %		
Individual Work	30 %		
Class Participation	10 %		

RE-SIT / RE-TAKE POLICY

Each student has four chances to pass this course distributed over two consecutive academic years: ordinary call exams and extraordinary call exams (re-sits) in June/July.

Students who do not comply with the 80% attendance rule during the semester will fail both calls for this Academic Year (ordinary and extraordinary) and have to re-take the course (i.e., re-enroll) in the next Academic Year.

- Students failing the course in the ordinary call (during the semester) will have to re-sit the exam in June / July (except those not complying with the attendance rule, who will not have that opportunity and must directly re-enroll in the course on the next Academic Year).
- The extraordinary call exams in June / July (re-sits) require your physical presence at the campus you are enrolled in (Segovia or Madrid). There is no possibility to change the date, location or format of any exam, under any circumstances. Dates and location of the June / July re-sit exams will be posted in advance. Please take this into consideration when planning your summer.
- The June / July re-sit exam will consist of a comprehensive exam. Your final grade for the course will depend on the performance in this exam only; continuous evaluation over the semester will not be taken into consideration. Students will have to achieve the minimum passing grade of 5 and can obtain a maximum grade of 8.0 (out of 10.0) – i.e., “notable” in the re-sit exam.
- Retakers: Students who failed the subject on a previous Academic Year and are now re-enrolled as re-takers in a course will be needed to check the syllabus of the assigned professor, as well as contact the professor individually, regarding the specific evaluation criteria for them as retakers in the course during that semester (ordinary call of that Academic Year). The maximum grade that may be obtained in the retake exam (3rd call) is 10.0.

After ordinary and extraordinary call exams are graded by the professor, you will have a possibility to attend a review session for that exam and course grade. Please be available to attend the session in order to clarify any concerns you might have regarding your exam. Your professor will inform you about the time and place of the review session. Any grade appeals require that the student attended the review session prior to appealing.

Students failing more than 18 ECTS credits after the June-July re-sits will be asked to leave the Program. Please, make sure to prepare yourself well for the exams in order to pass your failed subjects.

In case you decide to skip the opportunity to re-sit for an exam during the June / July extraordinary call, you will need to enroll in that course again for the next Academic Year as a re-taker and pay the corresponding extra cost. As you know, students have a total of four allowed calls to pass a given subject or course, in order to remain in the program

BIBLIOGRAPHY

Recommended

- Bill Lubanovic. *Introducing Python.* ISBN 9781492051343 (Digital)
- Svein Linge, Hans Petter Langtangen. *Programming for computations - Python: a gentle introduction to numerical simulations with Python 3..* ISBN 9783319324 (Digital)
- Eric Matthes. *Python Crash Course.* ISBN 9781593277390 (Digital)
- Travis E. Oliphant. *Guide to NumPy.* ISBN 9781517300074 (Digital)

WHEN QUESTIONS ARISE OUT OF CLASS

Email: If you have a question(s) that was not answered in class, you are welcome to ask your question(s) via email. I can be reached at: dprecioso@faculty.ie.edu

Although I will make every effort to respond to your question(s) as quickly and thoroughly as possible, please recognize that I may not be available when you send an email. Thus, please allow me up to 48 hours to respond before sending a follow-up email.

BEHAVIOR RULES

Please, check the University's Code of Conduct here. The Program Director may provide further indications.

ATTENDANCE POLICY

Please, check the University's Attendance Policy here. The Program Director may provide further indications.

ETHICAL POLICY

Please, check the University's Ethics Code here. The Program Director may provide further indications.