COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

COURSE SYLLABUS 2025-2026

GENERAL INFORMATION

Course information	
Name	Python Programming Fundamentals
Code	DTC-MIINT-517
Main program	Máster en Industria Inteligente
Offered in	Máster Universitario en Ingeniería Industrial + Máster en Industria Inteligente [1st year] Máster en Industria Inteligente [1st year]
Level	Master's Degree
Semester	1st (Fall)
Credits	1.5 ECTS
Type	Compulsory
Department	Telematics and Computer Science
Coordinator	Carlos Miguel Vallez Fernández

Instructor	
Name	Carlos Miguel Vallez Fernández
Department	Telematics and Computer Science
e-mail	cmvallez@comillas.edu
Office hours	Arrange an appointment through email.

COURSE SPECIFIC INFORMATION

Contextualization of the course

Contribution to the professional profile of the degree

Python has emerged as the leading programming language for numerous applications within Industry 4.0, particularly those driven by machine learning and artificial intelligence. This course aims to equip students with fundamental Python skills, essential for succeeding in subsequent subjects of the Master in Intelligent Industry program.

The course begins with an introduction to Python syntax, progresses through general and language-specific control and data structures, and concludes with the application of specialized libraries for data processing.

Prerequisites

Students willing to take this course should be familiar with undergraduate-level programming in any language (preferably C). Previous experience with Python is also desired although not strictly required.

Competences[1] – Objectives

Competences

General

CG1. Have acquired advanced knowledge and demonstrated, in a research and technological or highly specialized context, a detailed and well-founded understanding of the theoretical and practical aspects, as well as of the work methodology in one or more fields of study.

Haber adquirido conocimientos avanzados y demostrado, en un contexto de investigación científica y tecnológica o altamente especializado, una comprensión detallada y fundamentada de los aspectos teóricos y prácticos y de la metodología de trabajo en uno o más campos de estudio.

CG2. Know how to apply and integrate their knowledge, understanding, scientific rationale, and problem-solving skills to new and imprecisely defined environments, including highly specialized multidisciplinary research and professional contexts.

Saber aplicar e integrar sus conocimientos, la comprensión de estos, su fundamentación científica y sus capacidades de resolución de problemas en entornos nuevos y definidos de forma imprecisa, incluyendo contextos de carácter multidisciplinar tanto investigadores como profesionales altamente especializados.

CG7. Being able to take responsibility for their own professional development and their specialization in one or more fields of study.

Ser capaces de asumir la responsabilidad de su propio desarrollo profesional y de su especialización en uno o más campos de estudio.

Specific

CE3. Be able to design and implement software to solve industrial problems and critically evaluate its readability, maintainability, and efficiency.

Ser capaz de diseñar e implementar software para resolver problemas industriales y evaluar de manera crítica su legibilidad, mantenibilidad y eficiencia.

Learning outcomes

RA1. Create isolated development environments that include specific versions of all the dependencies required for a particular application.

RA2. Write code that follows PEP-8 conventions.

RA3. Manipulate text-based files using Python.

RA4. Handle errors and exceptions.

RA5. Understand the fundamentals of object-oriented programming, be able to implement new classes from scratch and use already existing ones.

RA6. Use the most popular data manipulation Python libraries at a basic level.

RA7. Interconnect applications following a RESTful architecture.

[1] Competences in English are a free translation of the official Spanish version.

CONTENTS

Contents
Unit 1. Introduction to Python development
1.1 Introduction 1.2 IDE setup and virtual environments 1.3 Basic syntax (reserved keywords, variable types, comments, docstrings, coding conventions) 1.4 My first program: “Hello, World” 1.5 Debugging
Unit 2. Control statements and strings
2.1 Conditionals 2.2 Loops 2.3 Strings
Unit 3. Data structures
3.1 Lists 3.2 Tuples 3.3 Dictionaries
Unit 4. Functions
4.1 Built-in functions 4.2 Modules 4.3 Programmer-defined functions 4.4 Type hints
Unit 5. Text files, exceptions and object-oriented programming
5.1 General file management concepts 5.2 Reading and writing text files 5.3 Exceptions 5.4 Introduction to object-oriented programming (OOP)
Unit 6. Data processing libraries
6.1 NumPy 6.2 Pandas 6.3 Matplotlib 6.4 REST APIs

TEACHING METHODOLOGY

General methodological aspects	
Each session will start with a brief explanation of the key concepts, followed by practical examples to reinforce learning.	
In-class activities	**Competences**
▪ **Lectures:** The lecturer will introduce the fundamental concepts of each unit, along with some practical recommendations, and will go through worked examples to support the explanation. Active participation will be encouraged by raising open questions to foster discussion.	CG1, CG7, CE3
▪ **Exercises:** Under the instructor's supervision, students will apply the concepts and techniques covered in the lectures to solve small coding problems.	CG1, CG6, CG7, CE3
▪ **Tutoring** for groups or individual students will be organized upon request.	–
Out-of-class activities	**Competences**
▪ Personal study of the course material and resolution of the proposed exercises.	CG1, CG7, CE3
▪ Development of a final project in small groups. If the student is taking the Data Modelling course simultaneously a single project will cover the requirements of both subjects.	CG1, CG2, CG7, CE3

STUDENT WORK-TIME SUMMARY

IN-CLASS HOURS	
Lectures	**Exercises**
7.5	7.5
OUT-OF-CLASS HOURS	
Self-study	**Final project**
10	20
ECTS credits:	**1.5 (45 hours)**

EVALUATION AND GRADING CRITERIA

The use of AI to produce entire assignments or significant parts of them, without citing the source or tool used, or without explicit permission in the assignment description, will be considered plagiarism and will be subject to the University's General Regulations.

Evaluation activities	Grading criteria	Weight
Classwork and participation	▪ Class exercises. ▪ Active participation during the lectures.	10%
Final exam	▪ Understanding of the general theoretical programming concepts and the particularities of the Python language. ▪ Application of these concepts to problem-solving though coding.	60%
Final project	▪ Application of theoretical concepts to real problem-solving. ▪ Quality of the proposed solution. ▪ Teamwork. ▪ Oral presentation skills. ▪ There will be an intra-group evaluation method to differentiate among team members.	30%

Grading

Regular assessment

- **Classwork and participation:** 10%
- **Final exam:** 60%
- **Final project:** 30%

In order to pass the course, the weighted average mark must be greater or equal to 5 out of 10 points, the mark of the final exam must be greater or equal to 5 out of 10 points, and the final project mark must be at least 5 out of 10 points. Otherwise, the final grade will be the lower of the three marks.

Retake

All students will take a final exam. The classwork and participation mark will be preserved, along with that of the final project only if it has a passing grade. Otherwise, a new project will have to be developed and defended publicly at the latest on the day of the retake exam. The final grade will be computed as in the regular assessment period and according to the same restrictions.

Course rules

- Class attendance is mandatory according to Article 93 of the General Regulations (Reglamento General) of Comillas Pontifical University and Article 6 of the Academic Rules (Normas Académicas) of the ICAI School of Engineering. Not complying with this requirement may have the following consequences:
  - Students who fail to attend more than 15% of the lectures may be denied the right to take the final exam during the regular assessment period.
  - Regarding laboratory, absence to more than 15% of the sessions can result in losing the right to take the final exam of the regular assessment period and the retake. Missed sessions must be made up for credit.
- Students who commit an irregularity in any graded activity will receive a mark of zero in the activity and disciplinary procedures will follow (cf. Article 168 of the General Regulations (Reglamento General) of Comillas Pontifical University).

Guidelines for the use of generative artificial intelligence (AI)

AI can be used to help complete the tasks related to classwork and the final project, including idea generation, writing, feedback, programming code hints/help, and evaluation. Students should critically analyze and modify the results suggested by the AI, demonstrating their understanding. In any case, the use of AI must be cited and the sources independently verified by the student. This means that students need to include in every submission the AI models or tools used, the prompts, the results obtained, and an explanation of the retrieved information or code. Failing to explain the code provided by the AI with the student's own words will be treated as plagiarism and handled according to the University's rules.

The use of AI is not permitted in any of the examination tests, nor in performance assessment tests.

WORK PLAN AND SCHEDULE

Activities	Date/Periodicity	Deadline
Class exercises	Every lecture	–
Review and self-study of the concepts covered in the lectures	After each lesson	–
Final exam	Two weeks after the last lecture	–
Final project	From the last lecture	The last week of the semester

BIBLIOGRAPHY AND RESOURCES

Basic references

- Slides prepared by the lecturer (available in Moodle).
- J. L. Gahete, C. M. Vallez, and A. Fernández-Pacheco, *Fundamentos de Programación en Python*, 1st Ed., McGraw-Hill, 2024, [In Spanish]. ISBN-13: 978-8-448-64527-4
- NumPy: The fundamental package for scientific computing with Python, [Online]. Available: https://numpy.org/
- Pandas: Fast, powerful, flexible and easy to use open source data analysis and manipulation tool, [Online]. Available: https://pandas.pydata.org/
- Matplotlib: Visualization with Python, [Online]. Available: https://matplotlib.org/

Complementary references

- M. Lutz, *Learning Python: Powerful Object-Oriented Programming*, 5th Ed., O'Reilly Media, 2013. ISBN-13: 978-1-449-35573-9
- M. Lutz, *Python Pocket Reference: Python in Your Pocket*, 5th Ed., O'Reilly Media, 2014. ISBN-13: 978-1-449-35701-6
- D. Beazley and B. Jones, *Python Cookbook: Recipes for Mastering Python 3*, 5th Ed., O'Reilly Media, 2013. ISBN-13: 978-1-449-34037-7
- E. Matthes, *Python Crash Course: A Hands-On, Project Based Introduction to Programming*, 3rd Ed., No Starch Press, 2023. ISBN-13: 978-1-718-50270-3
- D. Toomey, *Jupyter for Data Science: Exploratory analysis, statistical modeling, machine learning, and data visualization with Jupyter*, 1st Ed., Packt Publishing, 2017. ISBN-13: 978-1-785-88007-0
- R. Mitchell, *Web Scraping with Python: Data Extraction from the Modern Web*, O'Reilly Media, 3rd Ed., 2024. ISBN-13: 978-1-098-14535-4