



Syllabus

Academic Year	2022/2023
Program	Business Administration
course	Coding in action Labs (Module II)
Term	II semester
Year	1
SSD	ING-INF/05
Credits	6

INSTRUCTIONAL GOALS

This course builds upon the work done during the first module of the Coding in action Labs. It covers more advanced topics of programming such as object-oriented programming, computational cost and efficiency of a program, and so-called Pythonic programming. The course also introduces the main tools for data processing, analysis and visualization that are available in Python, which is one of the reference programming languages in the "data science" world.

INTENDED LEARNING OUTCOMES

They describe what a learner is expected to know, understand and be able to demonstrate after completion of a learning path.

Knowledge and understanding:

By the end of this course, the students will learn what object-oriented programming is and how this programming paradigm is implemented in Python. They will understand that a good program solve the given task making use of the least possible amount of resources. They will learn specific Python syntax and best practices needed to make programming faster and increase performance. Further, the students will learn how to use, at an introductory level, the most known Python packages for data analysis and visualization: NumPy, Pandas, Scikit-learn and Matplotlib.

Applying knowledge and understanding:

To test their understanding of the concepts seen in class, the students will be assigned projects that require to deal with real data. They will be asked to:

- Rapidly manipulate and process large data sets through dataframes and arrays
- Plot data and functions
- Operate basic descriptive analysis and modeling (statistics, histograms, interpolation, clustering, fitting)

The course will prepare the students for more advanced data analysis courses and make them ready to complete projects in other courses that require a computational approach to data.

Making judgements:

Upon completing the study program, students will be able to:

- Design algorithms in terms of objects that interact with one another
- Understand that an object can be considered an instance of a more general concept – a class – that defines the common features of all objects of that type
- Compare simple algorithms that solve the same task in terms of their computational cost
- Address simple data analysis project

Communications Skills:



Being introduced to the concepts of computational and memory efficiency and to their formalization, the students will understand that the “cost” of solving a problem can be precisely quantified and expressed.

Through examples, case studies and projects, the students will learn how to communicate the results of a data analysis task and how to justify the choice of specific algorithms, methods and techniques.

Learning skills:

The students will be introduced to a set of advanced/professional libraries for data analysis. At the end of the course, they will be able to autonomously browse the Python standard library as well as the web to find the right libraries and tools to perform a given task.

Pre-requisites	Having successfully completed the first module of the Coding in action Labs.
Course content	The course will cover the following aspects of computer programming: <ul style="list-style-type: none">• Classes and objects: classes, types, objects and instances; instances as return values; attributes and methods; objects mutability; instantiation and the <code>__init__</code> method; operator overloading and special methods; static methods and class methods; inheritance.• Pythonic programming: conditional expressions; EAFP (Easier to Ask for Forgiveness than Permission); list comprehension; generator expressions; any and all; sets.• Scientific programming: Numpy, arrays and broadcasting; Pandas, dataframes and series; Scikit-learn and basic machine learning with Python; Matplotlib and plotting in Python
Reference Books	Reference book (suggested but not mandatory): <ul style="list-style-type: none">• Allen B. Downey, “Think Python: How to Think Like a Computer Scientist (2nd Edition)”, O’Reilly, ISBN-13: 978-1491939369 Other (free) sources: <ul style="list-style-type: none">• Non-Programmer’s Tutorial for Python 3, available here.• The Python Tutorial, available here.
Teaching Methods	Traditional face-to-face teaching will be the main teaching method, but a few case studies will be considered that the students will have to address by themselves, alone or in groups, in line with a “learning by doing” paradigm. Students’ participation during lectures is strongly encouraged.
Assessment	Grading for this course will be based on a combination of programming assignments (50%) and a final exam (50%). The final exam will combine a multiple answer quiz with a programming problem. Both the programming assignments and the programming problem included in the final exam will be scored as follows: <ul style="list-style-type: none">• If the code produces the correct answer using the requested approach: 100%• If it generally uses the right approach, but the answer is incorrect due to some minor mistakes: 70-90%• If it correctly uses the core concepts and makes some progress in the attempt to solve the problem: 40-60%• If the answer demonstrates a lack of understanding of some or all of the core concepts: 0-40% At the teacher’s discretion, the students may be asked to provide a thorough description of their code.
