Libera Università Internazionale
degli Studi Sociali Guido Carli

LUISS

Syllabus

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

INSTRUCTIONAL GOALS	This course aims at providing an understanding of so-called "computational thinking" and a set of programming skills that are instrumental for data processing and analysis. Students will first learn how to use the command-line shell. They will then learn generic programming language concepts as they are implemented in Python. Python is a language that is simple to learn yet extremely flexible and powerful. Through an innovative teaching method, the students will learn how to tackle a programming problem without having a knowledgeable person to explain them how to solve it. They will learn to work in complete autonomy and to collaborate with other peers.
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**: The course will introduce the students to bash, a shell-specific scripting language, making them familiar with commands and their calling syntax. It will provide the students with an understanding of data structures and control flows, to then cover a few practical issues in statistical computing which include reading data, importing packages, writing functions, and debugging. By the end of this course, the students will gain a solid understanding of how to use the Python standard library to write programs, create functions and classes, write and access external modules. Besides the technical aspects, the special learning methods will teach students the basics of problem solving in a professional environment: how to manage a project, find resources needed to complete it, work in groups, ask for help or provide it to others. **Applying knowledge and understanding**: This course will teach students how to: • Use a command-line interface to explore a file system and access data • Use the main data types (strings, lists, dictionaries, and more) • Implement Control Flows (if-then statements, looping) • Organize code (functions, modules, packages) • Read and write data from local files (.txt, .csv, .json, etc) **Making judgements**: Upon completing the study program, students will be able to: • Decompose a problem into sub-problems that are easier to solve • Cooperate with other students to solve a programming problem by sharing the tasks • Design simple algorithms to solve common programming problems • Identify the most suitable data structure for the task at hand

	Communications Skills: To complete the course, the students will have to work in groups, collaborate with the community to overcome challenges, and share their experience with their peers. This will stimulate their communication skills with respect to all aspects of programming: describing a problem, discussing criticalities, asking for help, explaining an algorithm, commenting code. **Learning skills**: The innovative teaching method will force students to either find the answers by themselves or ask fellow students. In turn, they will often be asked to transmit notions to someone else. This method allows the students to sharpen their skills through research, experimentation and defending their personal approach to a given problem.
Pre-requisites	None.
Course content	The course will cover the following aspects of computer programming: • An introduction to programming: computer architectures; memory and data; CPU and programs; programming languages; problems, algorithms and programs. • The basics of command-line interfaces and of the bash scripting language. • How to use the Python interpreter: invoking the interpreter; argument passing; interactive mode; notebooks; online coding platforms. • Basic concepts of Python programming: variables and assignments; expressions and statements; operations; printing; comments; debbugging; data types; numbers and strings; input. • Functions: built-in functions; function calls; importing modules and functions; math functions; function composition; defining new functions; parameters and arguments; mandatory vs. optional arguments; arguments' order and keyword assignment; scope of a variable. • Taking decisions: boolean expressions and logical operators; conditional and alternative execution; if-elif-else statements; chained vs. nested conditionals. • Iterations: reassignment and updating variables; the while statement; the break statement; sequences and looping; the in operator; the for loop. • Data structures (strings, lists, tuples, dictionaries): definition, properties, operations and methods; indexing vs. assignment; mutability and immutability; map, filter and reduce; referencing and aliasing; packing and unpacking; lookup and reverse lookup; variable-length arguments. • Files: persistence; opening and closing and the with construct; reading and writing; format operator; filenames and paths; catching exceptions; pickling. • Modules and packages: defining a module; defining a package; importing a package vs. importing a module vs. importing a function; installing packages.
Reference Books	Reference book (suggested but not mandatory): • Allen B. Downey, "Think Python: How to Think Like a Computer Scientist (2nd Edition)", O'Reilly, ISBN-13: 978-1491939369 Other (free) sources: • Non-Programmer's Tutorial for Python 3, available here. • The Python Tutorial, available here.
Teaching Methods	The course will be organized in cooperation with the 42 programming school and will mostly embrace its innovative teaching methods. Face-to-face teaching will be replaced by a program based on problem solving. The students will have to carry out projects alone or in a group, and to discuss them with the teacher and with other students, who have not necessarily progressed at the same pace. The students will have to justify their choices, explaining why they decided to solve the project in a specific way. These moments help consolidate personal learning and share knowledge with the community. Collective intelligence, which allows good practices to emerge and groups to advance, plays a major role in this teaching method. The students are asked to think for themselves in front of a computer. In such a circumstance, failing

	and making mistakes will not be penalized, but rather considered part of a learning path. The pedagogical staff will be available to help the students find their own solutions.
Assessment	In line with the chosen teaching and learning methods, this course will not have a final exam. During the course, the students will be provided with a set of project that they have to complete in order to progress in the course and to successfully pass the exam.