



BASICS OF COMPUTING

INF/01 - 9 CFU - 1° semestre

Docente titolare dell'insegnamento

GIOVANNI GALLO

Email: gallo@dmi.unict.it

Edificio / Indirizzo: DMI, polo scinetifico di viale A.Doria(via s.Sofia

Telefono: 095 7383041

Orario ricevimento: mercoledì 9-12

OBIETTIVI FORMATIVI

Synthetic general description

The course introduces some fundamentals principles of Computer Scisnce but mostly focuses on an practical introduction to Programming using *Python* as reference language.

Focusing on the mosto common practices of Scientific computing imperative as well object oriented paradigm of programming will be considered. Some minor reference will be provided toward ad advanced use of spreadsheet computation models.

Elementary data structures (list, string, dataframes) will be introduced. Basic programing structures: conditional and case based control, iteration, recursion.

Functions and code factoring will be considered as well.

General educational objectives of teaching in terms of expected learning

1. **Knowledge and understanding:** the primary objective of the course is the students' acquisition of the "mind set" of programming, as well as an operative knowledge of *Python* and of some spreadsheet utilities.
2. **Applying knowledge and understanding:** it intends to provide the tools to achieve the following practical and professional skills:
 - To analyze computational problems and to code algorithmic ideas for their resolution;
 - To design, to describe, to implement and debug *Python* programs with professional tools;
 - To use built-in data structures for data management in scientific computing;
 - Understanding simple recursive algorithms;
 - To use specific libraries for scientific calculation;
 - To be able to read documentation of libraries.
 - To use, modify and create efficient and interactive spreadsheets.
3. **Making judgements:** through the examination of code examples and numerous practice exercises, the learner will be able, both independently and in a cooperative manner, to analyze

problems and design and implement related software solutions.

4. **Communication skills:** the student will acquire the necessary communication skills and expressive appropriateness in the use of technical verbal language in the context of computer programming.
5. **Learning skills:** the course aims to provide the learner with the necessary theoretical and practical methodologies to be used in professional contexts and, in particular, the ability to formulate and implement *ad-hoc* algorithms for solving new computational problems as well as the possibility of easily and quickly acquiring other programming languages.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

The course will include:

- a) frontal classes of mostly theoretical content (blackboard and slides) approx 10 hours;
- b) interactive laboratory classes with exercise and direct experiences (laptop projection, internet connection), students will be encouraged to use their own laptops (not mandatory) for approx 50 hours;

The classes will be held in presence (but different decisions of the authorities).

Hands-on learning and reference to typical case studies will be used through the full course.

PREREQUISITI RICHIESTI

No prerequisite is mandatory.

It will help a basic knowledge of spreadsheet usage and elementary notions of math and logic. Problem solving attitude and experience of rigorous quantitative thinking will be very helpful.

FREQUENZA LEZIONI

Requested according to the rules of the Corso di Studio

CONTENUTI DEL CORSO

1. Introduction to programming

1. The general structure of a digital computer.
2. Problems and Algorithms.
3. Flow chart, Structured linear notation, *Böhm-Jacopini* theorem.
4. Representation of information: integers and floating-point numbers, characters, strings.

2. Computational styles

1. "Turing-able" programming languages vs spreadsheets;
2. Basic cycle of computation in a spreadsheet: variables (value, format, dependencies)
3. Compilation and interpretation for traditional programming languages.
4. Installation of the development environment for the *Python* language. First program: Editing, Running, Debugging.

3. **Constructs of Python language**

1. Basic syntax, data types, predefined operators, I/O management.
2. Numbers and mathematical functions.
3. Flow control: constructs of selection and iterative.
4. Functions and recursion.

4. **Built-in data structures in Python**

1. Strings.
2. Lists, Tuples, Dictionaries.

5. **Object oriented programming in Python**

1. General ideas
2. Basic notation
3. Examples of simple objects and their usage;
4. Heredity

6. **Advanced topics**

1. Notable algorithms: Searching, Sorting, Merging. Basics of computational complexity.
2. Modules. Basic Python libraries for scientific computing and data analysis. Basic data visualization.

TESTI DI RIFERIMENTO

- 1) A.Downey, Think Python, 2nd Ed., Green Tea Press (online available).
- 2) M.Lutz, Learning Python, 4th Ed., O'Reilly (online available).
- 3) D.Pine, Introduction to Python for Science and Engineering, SMTEBooks - CRC Press (online available).
- 4) Jessen Havill - Discovering Computer Science: Interdisciplinary Problems, Principles, and Python Programming Chapman and Hall/CRC; 1 edizione (14 settembre 2015)

PROGRAMMAZIONE DEL CORSO

| Argomenti | Riferimenti testi |
|-----------|---|
| 1 | Introduction to computing and scientific computing |
| 2 | Spreadsheet model |
| 3 | Turing-able programming languages, interpretation and compilation |
| 4 | Python, installation and usage |
| 5 | Fundamental data types |
| 6 | Control: conditional, iteration, recursion |
| 7 | Basic data structures: list, string, tuples |
| 8 | Object oriented programming |
| 9 | Important algorithms |
| 10 | Modules and libraries: examples and usage |
| 11 | Using on line references |

VERIFICA DELL'APPRENDIMENTO

MODALITÀ DI VERIFICA DELL'APPRENDIMENTO

The verification will consist of:

- a programming assignment to be carried on in class on the exam date. It includes exercises in Python and using a spreadsheet (80% of the evaluation)
- an interactive discussion of the solution developed in the previous exercise (20% of the evaluation)

ESEMPI DI DOMANDE E/O ESERCIZI FREQUENTI

Part1

Given some data (in a list or dictionary format) extract information about them (elementary statistical descriptors, data cleaning, data aggregation tasks).

Part 2

Given some data in a spreadsheet extract information about them (elementary statistical descriptors, data cleaning, data aggregation tasks).

Part 3 (optional)

Write code for data visualization using Python graphics libraries
