UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Foundations of Computer Science

2627-1-FDS02Q009

Aims

At the end of the course, the students will understand how to query a database, and how to infer the implicit structure of a database from its tables.

Moreover, the students will be able to write and debug some simple programs in Python, to manage and analyze datasets consisting of a few interconnected tables, such as those usually available at kaggle.com (CSV, TSV, JSON formats). This part will taught with the Jupyter Notebook.

Finally, the students will be able to query a relational database using SQL, both via a dedicated interface and from a Python program.

Knowledge and understanding

This course provides basic knowledge and understanding on:

- Data bases
- SQL
- Programming in Python
- Jupyter notebooks
- Managing tabular datasets with Pandas

Applied Knowledge and Understanding

At the end of the course the students will be able to:

- Write SQL queries
- Write Python notebooks to manage datasets
- Write Python notebooks to analyze and clean datasets

Independent Judgment

Students will be able to independently assess the quality of data to be processed and perform exploratory data analysis using a Jupyter Notebook.

Communication Skills

Students will be able to describe the decision-making processes that guided their approach to analyzing and cleaning a dataset.

Learning Skills

Students will be able to learn new programming languages dedicated to statistical analysis, with a particular focus on the R language.

Contents

Organizing raw datasets: file system, delimited files.

Introduction to data bases. Relational Data Model and SQL. Select ... From … Where.

Introduction to programming with Python. Explorative programmaing. Managing tabular data.

Introduction to testing and debugging.

Detailed program

1. Organizing raw datasets
2. files, directories, types of files
3. main command-line commands
4. delimiter-separated values
5. Introduction to data bases.
6. The Relational Data Model.
7. SQL: Select ... From … Where on a table.
8. Querying two or more tables.
9. Introduction to programming in Python.
10. Arrays, lists, dictionaries
11. Loops
12. Organization of a programma: functions
13. Modules and libraries
14. Explorative programming. Managing tabular data.
15. The Jupyter Notebook
16. Pandas
17. Introduction to testing and debugging.

Prerequisites

Basic knowlege of any programming language

Teaching form

Lectures and exercises with a PC.
Lectures consist of two parts: the first part (approx. 20%) where the content will be introduced by the teacher, and a second (approx. 80%) , more interactive, part based on exercises.
Lectures will be neither recorded nor streamed.

Textbook and teaching resource

- Downey Think Python 2e: How To Think Like a Computer Scientist
- Downey Think Stats 2e
- Downey Elements of Data Science. This book is a bit too basic.
- VanderPlas, Python Data Science Handbook
- Elmasri, Navathe, Fundamentals of Database Systems

Semester

First (first 6 weeks only)

Assessment method

The exam consists of three parts: a written exam, a project to be developed in small groups (1 to 3 people), and an optional oral exam. All parts must be taken in English.

Written exam

It consists of four open-ended questions, each requiring an SQL query. Evaluation is based on the completeness and correctness of the answers. If the written exam is insufficient (a fail), the project cannot be presented.

Project (in Python)

The project integrates all the skills acquired throughout the course. The project specification is the same for all exam sessions until September 2026.

Submission guidelines: Following the successful completion of the written exam, and in any case within one month of its date (and at least 3 days before the oral exam, if taken), the project must be publicly uploaded to GitHub as a single Jupyter notebook. The link to the repository must be submitted via a dedicated online form.

Use of LLMs/Generative AI: This is permitted, but the notebook must include a dedicated section describing how these tools were used. The project is evaluated on correctness, completeness, and originality (especially compared to standard AI-generated solutions).

If the discussion reveals undisclosed uses of AI, a new project will be assigned and the written exam will be voided.

Oral exam

It must be taken within one month of the written exam and consists of a few questions on the project and all topics covered in class, aimed at assessing your understanding of the course syllabus.
Students can choose to answer questions only on the project (not on the topics covered in class): in this case the maximum grade is 24/30.
An insufficient grade in the oral exam results in a fail for the entire exam.

Final Grading

The baseline grade is calculated using the geometric mean (rounded down) of the written exam grade and the grades of the three components of the project (each expressed on a scale from 0 to 30). The oral exam modifies this grade by a maximum of 3 points, either upwards or downwards.

Registration for the written exam via the online student registry (segreteria online) is mandatory. No exceptions will be made.

Office hours

Office hourse are online. You can book a meeting at https://www.unimib.it/gianluca-della-vedova

Sustainable Development Goals

QUALITY EDUCATION