UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Foundations of Computer Science

2526-1-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

Ability to apply 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 evaluation

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

Assessment method

The exam consists of 2 parts: a written exam, and an oral discussion of a project completed in small teams (from 1 to 3 students). Both parts must be completed in English.

The written exam has open questions on SQL queries and on the Python basics.

The project covers the course contents.

The grading of the written exam is based on the correctness and the completeness of the answers. The written exam consists of writing 4 SQL queries, where each query corresponds to an open-ended question, and 2 questions on the syntax and semantics of a short Python program.

The grading of the project is based on the individual contribution that is displayed during the a complete and thorough oral discussion, including the ability to explain the choices made to complete the project, and on the fitness of the project to perform the required analysis. Both the written exam and the project discussion must be completed by September 2026.
No later than 24 hours before the project discussion, the project must be available on github. The link to the github repo must be provided when booking the time slot for the discussion.
If you don't show a complete understanding of your project, you will be assigned a different project and you will take to take again the written exam.

The final grade is computed with weights 50% the written exam and 50% the project.

There are no in-progress exams. Beware that you must be registered via "segreterie online" to take the exam. If you are not registered, you will not allowed to take the exam. 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