# Database systems and computer programming

## Prof. Jianyi Lin

***COURSE AIMS AND INTENDED LEARNING OUTCOMES***

The increasing application of computational methods for data analysis requires a thorough understanding of the technical tools to devise effective software for handling data. This course aims at providing students with solid skills in the principles of computer programming through the modern Python language, as well as an introduction to the formal and technological framework for modelling, storing and extracting information with database systems in the well-established relational model. Throughout the course, the analysis of applications and examples will foster the computational problem solving competence in data representation and processing.

At the end of the course, the student is expected to achieve the following course learning outcomes:

– understand the fundamental constructs of computer programming under the structured paradigm, such as flow control structures and functions; demonstrate knowledge of relational data model as well as SQL key concepts (*Knowledge and understanding*);

– understand, elaborate and implement programming and database solutions for the given requirement specifications in a particular language; improve problem solving skills that demand computational thinking (*Applying knowledge and understanding*);

– recognise, distinguish and choose good coding practices and more effective implementations in programming and database operations; identify elements in the relational data model which are appropriate for describing facts (*Making judgements*):

– express the purpose of some code using computer science terminology; properly justify the implementation and outline the program or data architecture with a basic engineering approach (*Communication skills*);

– autonomously refer to authoritative documentation of the language and framework adopted; recognise language/formalism independent constructs by abstracting from their concrete usage (*Lifelong learning skills*).

***COURSE CONTENT***

1. Introduction to modern computer architecture and programming.

2. Python environment and simple programs.

3. Fundamental constructs in structured programming: expressions, variables & data types, input & output, conditional branching, definite (for) loops, conditional (while) loops.

4. Functions: definition, arguments, variable scope.

5. Sequences, dictionaries and objects.

6. Applications to mathematical and textual examples, first libraries for data science (NumPy, SciPy, Pandas), recursion in programming.

7. Introduction to database systems and relational data model: relations, tuples, attributes, keys.

8. Relational algebra: selection, projection, Cartesian product, set operations, join operations, renaming.

9. Structured Query Language: basic syntax, where-clause, set operations, sorting, aggregate functions, join operations

10. Database management: table creation & deletion, constraints, programming interface to DBMS

11. Enterprise topics: data warehouse for analytics, entity-relationship design model

***READING LIST[[1]](#footnote-1)***

J. M. Zelle. Python programming: an introduction to computer science, 3rd Edition. Franklin, Beedle, 2016.

J. S. Conery. Explorations in Computing: An Introduction to Computer Science and Python Programming. Chapman and Hall/CRC, 2014.

A. Silberschatz, H. F. Korth, S. Sudarshan. Database System Concepts, 7th Edition. McGraw-Hill, 2019.

Further instructional material, e.g. class notes, source code, handouts, will be posted on Blackboard.

***TEACHING METHOD***

A blend of lectures (52 hours) and practice sessions in computer lab (8 hours).

***ASSESSMENT METHOD AND CRITERIA***

Written midterm exam (40% weight) and written final exam (40% weight) with closed-ended and open-ended questions on concepts and languages. A practical assessment focusing on a small project meant to test the ability in coding and database development at the end of the course (20% weight).

***NOTES AND PREREQUISITES***

Students are supposed to have good general skills in using a computer, such as skills in managing files, searching and navigating the Internet, using text editing and spreadsheet applications, but there are no prerequisites concerning programming languages or database software.

In case the current Covid-19 health emergency does not allow frontal teaching, remote teaching will be carried out following procedures that will be promptly notified to students.

1. I testi indicati nella bibliografia sono acquistabili presso le librerie di Ateneo; è possibile acquistarli anche presso altri rivenditori. [↑](#footnote-ref-1)