UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

# COURSE SYLLABUS

# Elements of Computer Science

**2627-1-F9202P009**

## Aims

The first aim is to provide students with the basic understandings regarding the representation, processing, transmission and communication of digital information using a computer or a network of computers. The second aim is to teach students, thanks to theoretical lessons and practical exercises, the basic skills of computer programming in the Python field.

Learning outcomes as set out in the Dublin Descriptors:

Knowledge and understanding
Theoretical foundations: to be familiar with the basic principles of computer science, including the concepts of algorithms, computation and data structures.
Computer architecture: to understand how a computer works at a low level (CPU, memory, input/output) and the interaction between hardware and software.
Programming languages: to have a basic knowledge of at least one programming language (e.g. Python) and its fundamental constructs (variables, loops, conditionals, functions).
Operating systems: to be familiar with the basic concepts of operating systems, such as processes, memory, file systems and resource management.
Computer networks: understanding the principles of networking, including basic protocols (TCP/IP, HTTP), IP addressing and how the Internet works.
Databases: understanding the fundamental concepts of relational databases (tables, keys, SQL queries) and their practical application.

Ability to apply knowledge and understanding
Programming: being able to understand simple programmes in a programming language and solve basic problems.
Data management: being able to create, modify and query a simple database using SQL.
Technical problem-solving: being able to identify and resolve common problems relating to hardware, software or networks.

Independent judgement

Evaluation of solutions: being able to critically evaluate different solutions to an IT problem, taking into account efficiency, readability and maintainability.
Choice of tools: being able to select the most suitable software and hardware tools for a specific task.
Ethics and responsibility: reflecting on the ethical and social implications of IT, such as privacy, data security and accessibility.

Communication skills
Documentation: being able to write clear and comprehensible documentation for the code, projects or systems developed.
Teamwork: collaborating effectively within a team, communicating technical ideas in a way that is accessible even to non-experts.
Technical communication: explaining IT concepts to people with different levels of expertise, avoiding excessive jargon.

Learning ability
Self-learning: Being able to learn new languages, tools or technologies independently, using online resources, official documentation and tutorials.
Adaptability: Adapting quickly to new technologies, programming languages or development methodologies.
Problem-solving: Knowing how to search for and evaluate solutions to technical problems using resources such as Stack Overflow, specialist forums or official documentation.
Continuous professional development: Keeping up to date with trends and developments in the field of IT by following blogs, attending conferences or taking online courses.

## Contents

The teaching is made of two main parts. A first part, mainly theoretical, which deals with the fundamental concepts of: digital representation of information in computers, information systems and systems for knowledge management, computer architecture, operating systems, computer networks and internet of things.

A second part, practical-theoretical, introduces the main concepts of coding, by showing some Python examples. This part will be supported by an *ad hoc* exercise activity.

## Detailed program

Digital representation of information: definition of information, digital representation of numbers, audio, images, video and characters, knowledge representation in a computer.

Information processing systems: the computer machine, types of computers, operating systems.

Computer networks and internet: computer networks basics, main types of networks, transmission media and main network devices, distributed multimedia applications.

Computer programming: definition of algorithm, programming language paradigms, programs, data types, basic data structures and programming constructs.

Examples of programming using Python and perhaps other programming languages.

## Prerequisites

None

## Teaching form

A part of lectures will be theoretical and will take place in a classroom; another part of the lectures will be exercise classes that will take place in a computational laboratory and/or in the classrom. For these exercise classes, students will have to use a personal computer (their own or university's ones). The exercise classes will be aimed at the development of simple script applications in the Python programming language.

We foresee the sharing the of all the didactics content needed for sudying these topics and for preparing the exam tests through eLearning tools.

All lectures and teaching sessions will take place in person, except in exceptional circumstances where specific problems arise.
The course will consist of:

- 30 hours of lectures (lecture-based teaching)
- 24 hours of laboratory sessions (interactive teaching)
- approximately 15 hours of tutorial sessions (interactive teaching)

## Textbook and teaching resource

Handouts provided by the teachers (alternative textbooks will be suggested to students not speaking Italian)

Plotting and programming in Python online free tutorial by Software Carpentry (in English)

## Semester

First semester

## Assessment method

A written exam consisting of multiple-choice and open-ended questions to assess the knowledge acquired on the course topics. The exam also includes several exercises to assess students' understanding and ability to apply the key concepts learned.

An optional supplementary oral exam, in which the oral exam may confirm the written exam grade, add 1 or 2 points to it, or subtract 1 or 2 points from it.

Translated with DeepL.com (free version)

## Office hours

Through appointment (writing to davide.chicco@unimib.it )

## Sustainable Development Goals

QUALITY EDUCATION