UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

# SYLLABUS DEL CORSO

## Laboratorio di Matematica e Informatica

**2223-1-E3501Q066**

### Aims

This course aims at introducing the basic *knowledge* of computer systems architecture and networks, as well as different programming pardigms. Moreover, the course will provide *competencies* to identify algorithms to solve simple problems and implementing them into the Java Programming Language, according to the imperative programming paradigm.

### Contents

Von Neumann's Model of Calculators. Components and functionalities of operating systems. Introduction to Computer Neworks. Programming Languages. Structured Programming in Java.

### Detailed program

Architecture of Calculators

- The Von Neumann model and basic notions on information representation
- Introduction to Operating Sytstems
- Basic notions of Computer Networks

Structured Programming in Java

- Programming languages and translators taxonomy
- The Java Virtual Machine
- Algorithms and programs

- Primitive Data types in Java.
- Flow Control in Java
- Arrays of Primitiva Data Types
- Methods in Java: definition and invocation
- Introduction to recursive algorithm design and implementation

## Prerequisites

Nothing

## Teaching form

- Lessons, 4 credits
- Laboratory, 2 credits

## Textbook and teaching resource

All the information about the course as well as the lessons slides and practical exercises will be available through the learning platform of the University, at the elearning.unimib.it link.

The suggested texdtbook will be:

W. Savitch: "Programmazione di base e avanzata con Java", a cura di Daniela Micucci, 2nd edition, Pearson

## Semester

Second semester

## Assessment method

### Examination type

Written and Oral examination; the oral examination is not mandatory, but necessary to obtain a "cum laude" merit. The mark range is 18-30/30. The oral examination is about both theoretical questions and practical exercises and can increase the result of written examination by at most 4 points.

The written examination is divided into two parts: the first one is devoted to evaluate theoretical skills aobut structured programming, by means of a collection of close-ended questions; the second one concerns the design and implementation of a simple software program, with the aim to demonstrate the student's capability to solve correctly a simple practical problem, on the basis of programming principles considered during the course, without

generating any kind of error (i.e. compile time, runtime, logical errors).

The arithmetic mean (possibly weighted) of the two marks defines the final mark proposed to the student: in case it is sufficient, the student can accept it as is or modify it by means of an oral examination (possibly decreasing the final mark). Oral examination is possible if and only if written examination is sufficient. The teacher has the faculty to establish mandatory oral examinations for those students whose written examinations, although sufficient, present some criticalities: for example, in case of not sufficient theoretical questions whereas pratical exercises are good, or viceversa.

Five exam sessions are stated: June, July, September, January and February; moreover two partial examinations are proposed to studets during the course. Partial examinations can be attended if and only if the student has accomplished the tasks proposed during practical laboratories.

## Office hours

Thursday, between 11 a.m. and 12 a.m., or by appointment.

## Sustainable Development Goals