UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Algoritmi e Programmazione

2627-2-E3502Q013

Aims

The primary objective of the course is to equip students with the skills, practical abilities, and problem-solving strategies for designing and implementing software systems using the object-oriented programming paradigm.
To achieve these objectives, students will be provided with the necessary skills for modeling simple domains using the UML language and coding them using the object-oriented programming paradigm using the Java programming language.
Upon completion of the course, the student will:
• know the principles of modeling, design, and implementation of software systems according to the object-oriented programming paradigm (**Knowledge and Understanding**)
• be able to apply this knowledge to the modeling of simple domains using the UML language and their implementation in Java, following the principles of object-oriented programming (**Applying Knowledge and Understanding**)
• be able to integrate various problem-solving strategies to critically address design and implementation choices in an object-oriented software context (**Making Judgments**)
• be able to justify the design solutions adopted, correctly using the technical vocabulary of object-oriented programming and modeling tools (**Communication Skills**)
• have learned how to independently acquire additional techniques and tools related to the design and development of object-oriented software (**Learning Skills**)

Contents

Modeling and programming using the object-oriented paradigm and an introduction to software design. By the end of the course, students will be able to model a problem using the object paradigm and encode it in a software program written in an object-oriented language. The reference language is Java.

Detailed program

- Introduction to fundamental concepts of the object-oriented programming paradigm (encapsulation, inheritance, polymorphism)
- Introduction to UML (Unified Modeling Language)
- Overview of the software life cycle
- Java as a language and as a platform
- Fundamentals elements of object-oriented paradigm in Java context: classes and objects, attributes and methods
- Advanced elements of object-oriented paradigm in Java context: inheritance and polymorphism
- Exceptions, ArrayList, Generics

Prerequisites

It is not verified that the student has passed the Mathematics and Computer Science Laboratory exam (first year), but it is a prerequisite for understanding the contents of this course to have knowledge of the contents covered therein, in particular of **structured programming with the Java programming language**

Teaching form

- Lecture, 4 credits
- Classroom exercises, 1 credit
- Classroom lab, 1 credit

The course uses a hybrid teaching approach that combines classroom teaching (DE) and interactive teaching (DI). DE includes the presentation and detailed explanation of theoretical content. DI involves active student participation, both through questions and requests during in-person practical exercises and through exercises and problems solved during practical labs, supported by a tutor/tutor.
Lectures are held in person and are taught in Italian, for 32 hours (4 credits).
Classroom exercises are held in person and are taught in Italian, for 12 hours (1 credit).
Laboratories are taught remotely and are taught in Italian, for 12 hours (1 credit).

Textbook and teaching resource

All course information, lecture slides, and laboratory exercises will be available on the University e-learning platform [https://elearning.unimib.it].

Reference text for the course:

Programmazione di base e avanzata con Java 3/Ed.
Walter Savitch, Daniela Micucci
ISBN Cartaceo: 9788891916020 – ISBN Digitale: 9788891916037
Link alla pagina di catalogo: https://he.pearson.it/bundle/818?isbn=9788891916037

Specifically, the content of chapters 8-13 (as a continuation of the content of chapters 1-7, covered in the Mathematics and Computer Science Laboratory course, first year)

Semester

Second semester

Assessment method

Written exam and optional oral exam (upon request by students who have obtained at least a 26/30 grade in the written exam). Grades will be awarded out of 30 (18-30 cum laude). The oral exam, which may include both theoretical questions and practical programming exercises, carries less weight than the written exam, resulting in a maximum increase of 4 points over the written exam score.
The written exam is divided into two parts: the first, through a series of closed-ended questions, assesses knowledge of the theoretical foundations of object-oriented programming; the second, through the implementation of a simple software system, assesses the student's ability to practically create a program capable of correctly solving a simple application problem, meeting the specifics of the problem at hand and adhering to the principles of object-oriented programming presented in class, without generating errors (compilation, runtime, or logic).
The instructor reserves the right to require additional investigation, through a mandatory oral exam, in cases where the written exam, while rated sufficient, presents critical issues: for example, a failure in the theoretical part but a very satisfactory practical part, or vice versa.
Five exam sessions are scheduled throughout the year, according to the program schedule. Additionally, two partial written exams will be held during the course. If the overall result is positive, the grade will be recorded at the end of the course (sixth exam).

Office hours

Student reception by appointment to be agreed by sending an email

Sustainable Development Goals