UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Laboratorio di Matematica e Informatica

2627-1-E3502Q005

Aims

Knowledge and understanding
Upon completion of the course, students will have acquired fundamental knowledge regarding computer systems architecture and computer networks, as well as a critical understanding of the main programming paradigms, with particular focus on the imperative paradigm.
Applying knowledge and understanding
Students will be able to analyze computational problems of elementary complexity, design appropriate solution algorithms, and implement them effectively using the Java programming language, demonstrating mastery of imperative programming techniques.
Making judgements
Students will develop the ability to critically evaluate the appropriateness of different algorithmic solutions, compare the efficiency of alternative approaches, and select the most suitable programming strategies in relation to the context of the problem to be solved.
Communication skills
Students will be able to document their code clearly and professionally, explain the implementation choices adopted, and effectively communicate the results of their programming work using appropriate technical language.
Learning skills
Students will acquire the methodological competencies necessary to independently deepen their knowledge of new programming languages, computer technologies, and software development paradigms, building the foundation for continuous learning in the field of computer science applied to mathematics.
Sustainable development goals
Quality education

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

A hybrid teaching approach is used, that combines lecture-based teaching (DE) and interactive teaching (DI). DE involves detailed presentation and explanation of theoretical content. DI includes active student participation through exercises and problems, to be accomplished during the practical laboratories under the supervision of a tutor.
Lessons (32 hours) are conducted in person and are delivered in Italian.
Laboratories (24 hours) are conducted at distance and are in Italian.

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, 3rd 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, equal or grater than 26/30. 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.

Six exam sessions are stated: June, July, September, January and February; moreover two partial examinations are proposed to students during the course. (6th exam session).

Office hours

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

Sustainable Development Goals

QUALITY EDUCATION