## PROGRAMMAZIONE II

### FRANCESCA LEVI

<table>
<thead>
<tr>
<th>Anno accademico</th>
<th>2020/21</th>
</tr>
</thead>
<tbody>
<tr>
<td>CdS</td>
<td>INFORMATICA</td>
</tr>
<tr>
<td>Codice</td>
<td>273AA</td>
</tr>
<tr>
<td>CFU</td>
<td>9</td>
</tr>
</tbody>
</table>

### Obiettivi di apprendimento

**Conoscenze**

The student who successfully completes the course will have the ability to design and verify advanced programming abstractions using the abstraction mechanism provided by object-oriented programming languages like Java. He or she will have a good understanding of the basic components of the run-time supports of high-level programming languages.

**Modalità di verifica delle conoscenze**

During the written exam the student must demonstrate to be able to put into practice some notions of programming language semantics learned during the course. The student's ability to explain correctly the main topics presented during the course at the board will be assessed during the oral exam.

Methods:

- Final oral exam
- Final written exam
- Periodic written tests


Resta la discussione del progetto.

### Indicazioni metodologiche

**Delivery:** face to face

**Learning activities:**

- attending lectures
- participation in discussions
- individual study

**Attendance:** Advised

**Teaching methods:**

- Lectures
- project work

### Programma (contenuti dell'insegnamento)

Programming Language Paradigms. Run-time structures: environment, memory, heap; blocks, subprograms, activation records, stack; objects; interpretation, compilation. Object-Oriented Programming. The kernel of the Java language: classes, objects, inheritance, data abstraction (specification and implementation, verification techniques based on induction), abstraction through hierarchies, Java polymorphism. Concurrent programming techniques.

### Bibliografia e materiale didattico

Ultimo aggiornamento 07/08/2020 13:00