



Programme of Course "Object-Oriented Software Design"

- Code: DT0208
- Type of course unit: Compulsory (Bachelor Degree in Computer Science curriculum General)
- Level of course unit: Undergraduate Degrees
- Semester: 2

Number of ects credits: (Bachelor Degree in Computer Science) 6 (workload 150 hours)

Teachers: Romina Eramo (romina.eramo@di.univaq.it)

1	<b>Course objectives</b>	The focus of the course is on achieving advanced knowledge of the Object-Oriented Programming paradigm and the Object-Oriented Software Engineering, and experience with the JAVA language.
2	<b>Course content and learning outcomes (dublin descriptors)</b>	<p>Topics of the module include:</p> <ul style="list-style-type: none"> <li>• Introduction to the Object-Oriented paradigm</li> <li>• Introduction to the Object-Oriented Software Engineering</li> <li>• Requirements definition, systems architecture design, software design</li> <li>• Java: Classes, objects, inheritance, collections, interfaces and exceptions.</li> <li>• Advanced aspects of JAVA: execution, documentation, threads, I/O, AWT, JDBC</li> <li>• Design Patterns</li> </ul> <p>On successful completion of this module, the student should :</p> <ul style="list-style-type: none"> <li>• To have solid knowledge of methods and techniques in Object Oriented Programming (OOP).</li> </ul> <p>To understand the fundamental OOP concepts of Objects and their usage, Class design, Interfaces, Relationships between Classes, Inheritance, Polymorphism. To applying acquired knowledge to Java programming.</p> <p>To provide a description of a problem and to design a first solution by: -defining the requirements of the system; - performing an analytical description of the system architecture, relevant entities and relationships in an application domain; and finally -designing the software of the system. To demonstrate skill in OO design to propose and communicate solutions. To show skills in programming also through the use of tools like IDEs.</p> <p>To exploit the acquired knowledge and abilities to solve problems in a larger variety of contexts. To demonstrate the capacity for reading and understand other texts on related topics.</p>
3	<b>Course prerequisites</b>	Basic notions of imperative and object-oriented programming.
4	<b>Teaching methods and language</b>	<p><b>Language:</b> Italian</p> <p><b>Reference textbooks</b></p> <ul style="list-style-type: none"> <li>• Gamma, Helm, Johnson, Vlissides (GoF), <i>Design Patterns: Elements of Reusable Object-Oriented Software</i> . Addison-Wesley.</li> <li>• Bruce Eckel, <i>Thinking in Java 4 ed. - Concorrenza e interfacce grafiche</i>. Pearson. (vol. 3)</li> <li>• Bruce Eckel, <i>Thinking in Java 4 ed. - I fondamenti</i>. Pearson. (vol. 1) .</li> <li>• Bruce Eckel, <i>Thinking in Java 4 ed. - Tecniche avanzate</i>. Pearson. (vol. 2)</li> </ul>
5	<b>Assessment methods</b>	The exam consists of a project to be developed in groups of at most four students, with its oral discussion.