# SYLLABUS
## COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>COURSE TITLE</th>
<th>Programming Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>COURSE CODE</td>
<td>76204</td>
</tr>
<tr>
<td>SCIENTIFIC SECTOR</td>
<td>INF/01</td>
</tr>
<tr>
<td>DEGREE</td>
<td>Bachelor in Computer Science</td>
</tr>
<tr>
<td>SEMESTER</td>
<td>2nd</td>
</tr>
<tr>
<td>YEAR</td>
<td>1st</td>
</tr>
<tr>
<td>CREDITS</td>
<td>9</td>
</tr>
<tr>
<td>TOTAL LECTURING HOURS</td>
<td>60</td>
</tr>
<tr>
<td>TOTAL LAB HOURS</td>
<td>30</td>
</tr>
<tr>
<td>PREREQUISITES</td>
<td>Students should be familiar with the basic knowledge of object-oriented programming and Java, as taught in the course “Computer Programming”</td>
</tr>
<tr>
<td>COURSE PAGE</td>
<td><a href="https://ole.unibz.it/course/view.php?id=4743">https://ole.unibz.it/course/view.php?id=4743</a></td>
</tr>
</tbody>
</table>

**SPECIFIC EDUCATIONAL OBJECTIVES**
Type of course: “caratterizzanti” for L-31  
Scientific area: “Discipline informatiche” for L-31  

The course is designed to give specific professional skills. It will provide students with advanced techniques in Java. In particular, students will acquire knowledge in the overall architecture and components of Java SDK and JRE, use of advanced programming techniques (e.g. multi-threads, reading/writing streams, generics, regular expressions, exception handling and testing) and code documentation (e.g. generate API documentation) of the software developed.

<table>
<thead>
<tr>
<th>LECTURER</th>
<th>Marko Tkalcic, office POS 1.13, <a href="mailto:marko.tkalcic@unibz.it">marko.tkalcic@unibz.it</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>SCIENTIFIC SECTOR OF THE LECTURER</td>
<td>INF/01</td>
</tr>
<tr>
<td>TEACHING LANGUAGE</td>
<td>English</td>
</tr>
<tr>
<td>OFFICE HOURS</td>
<td>By appointment, office POS 1.13</td>
</tr>
<tr>
<td>TEACHING ASSISTANT</td>
<td>Billero Riccardo, Cardoso Evellin</td>
</tr>
</tbody>
</table>
## OFFICE HOURS
TBA

## LIST OF TOPICS COVERED
- Memory models in Java
- Virtual functions, late binding, overriding, and overloading
- Exception handling
- Reflection and runtime type identification
- Generics and collections
- I/O, serialization and XML/JSON processing
- Designing large applications: design patterns
- Multithreading
- Code optimization

## TEACHING FORMAT
- Frontal lectures
- Lab exercises
- Individual projects

## LEARNING OUTCOMES

### Knowledge and understanding
- Know basic and advanced programming techniques
- Have a basic knowledge of the most important data structures and their use in programming languages

### Applying knowledge and understanding
- Be able to develop small and medium size programs using Java
- Be able to solve problems through the application of programming methods

### Making judgments
- Be able to work autonomously according to the own level of knowledge and understanding

### Communication skills
- Be able to structure and write scientific documentation

### Learning skills
- Have acquired learning capabilities to pursue further studies with a high degree of autonomy

## ASSESSMENT
The assessment is based on the lab assessment and the written exam. The lab assessment is composed of weekly assignments and a project. The weekly assignments are optional and motivate the students to study throughout the semester. The project is mandatory and evaluates the students' abilities to integrate the learned techniques into a stand-alone software project.

The written exam evaluates the students' understanding of the theoretical backgrounds and solving smaller, individual programming tasks.

## ASSESSMENT LANGUAGE
English

## EVALUATION CRITERIA AND CRITERIA FOR AWARDING MARKS

The assessment is based on (i) the lab assessment (up to 30 points) and (ii) the written exam (up to 30 points).

The lab assessment consists of (i.i) weekly assignments (optional, up to 10 points) and (i.ii) a project (mandatory, up to 20 points).

The final mark is: (a) the score of the written exam if that is higher than the lab assessment score, or (b) the average between the lab assessment
score and the written exam score if the lab score is higher than the written exam score.

In order to be eligible to enroll for the written exam, the student needs to obtain at least 5 points (out of 20) from the project. The lab assessment is a sum of the scores from the weekly assignments and the project score. The weekly assignments scores can be obtained only during the lectures period. The project must be submitted before each written exam (deadlines to be defined depending on the exam dates).

There is a mid-term exam. It is optional. The mid-term exam accounts for the first part (50%) of the final written exam. The grade of the mid-term exam is valid for all three regular exam sessions.

**REQUIRED READINGS**

Lecture notes will be handed out during the course.

**SUPPLEMENTARY READINGS**


**SOFTWARE USED**

- Eclipse IDE
- JDK