

## SYLLABUS COURSE DESCRIPTION

<b>COURSE TITLE</b>	<b>Programming Project</b>
<b>COURSE CODE</b>	76204
<b>SCIENTIFIC SECTOR</b>	INF/01
<b>DEGREE</b>	Bachelor in Computer Science
<b>SEMESTER</b>	2nd
<b>YEAR</b>	1st
<b>CREDITS</b>	9
<b>TOTAL LECTURING HOURS</b>	60
<b>TOTAL LAB HOURS</b>	30
<b>PREREQUISITES</b>	Students should be familiar with the basic knowledge of object oriented programming and Java, as taught in the course "Computer Programming"
<b>COURSE PAGE</b>	<a href="http://www.inf.unibz.it/~tkalcic/teaching_pp2017-2018.html">http://www.inf.unibz.it/~tkalcic/teaching_pp2017-2018.html</a>
<b>SPECIFIC EDUCATIONAL OBJECTIVES</b>	<ul style="list-style-type: none"> <li>• Type of course: "caratterizzanti" for L-31</li> <li>• Scientific area: "Discipline informatiche" for L-31</li> </ul> <p>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.</p>
<b>LECTURER</b>	Marko Tkalcić, office POS 1.13 marko.tkalcić@unibz.it <a href="http://www.inf.unibz.it/~tkalcic/">http://www.inf.unibz.it/~tkalcic/</a>
<b>SCIENTIFIC SECTOR OF THE LECTURER</b>	INF/01
<b>TEACHING LANGUAGE</b>	English
<b>OFFICE HOURS</b>	To be defined By appointment, office POS 1.13, marko.tkalcić@unibz.it
<b>TEACHING</b>	TBA

<b>ASSISTANT</b>	
<b>OFFICE HOURS</b>	TBA
<b>LIST OF TOPICS COVERED</b>	<ul style="list-style-type: none"> <li>○ Memory models in Java</li> <li>○ Virtual functions, late binding, overriding, and overloading</li> <li>○ Exception handling</li> <li>○ Reflection and runtime type identification</li> <li>○ Generics and collections</li> <li>○ I/O, serialization and XML/JSON processing</li> <li>○ Designing large applications: design patterns</li> <li>○ Multithreading</li> <li>○ Code optimization</li> </ul>
<b>TEACHING FORMAT</b>	<p>Frontal lectures          Lab exercises          Individual projects</p>

<b>LEARNING OUTCOMES</b>	<p><b>Knowledge and understanding</b></p> <ul style="list-style-type: none"> <li>• Have a deep knowledge of the main programming techniques</li> <li>• Knowledge of the most important data structures and their use in programming languages;</li> <li>• Knowledge the fundamental principles of programming;</li> <li>• Understand the key principles, the structures and the organization of computer systems</li> </ul> <p><b>Applying knowledge and understanding</b></p> <ul style="list-style-type: none"> <li>• Be able to develop small and medium size programs using different programming languages and paradigms;</li> <li>• Be able to solve problems through the application of programming methods;</li> <li>• Be able to develop programs, which interact with the operating system of a modern computer;</li> <li>• Be able to adopt the own knowledge in new application areas such as web engineering or business and IT engineering.</li> </ul> <p><b>Making judgments</b></p> <ul style="list-style-type: none"> <li>• Be able to work autonomously according to the own level of knowledge and understanding and to take the responsibility for development projects or IT consulting;</li> </ul> <p><b>Communication skills</b></p> <ul style="list-style-type: none"> <li>• Be able to structure and write scientific documentation</li> </ul> <p><b>Learning skills</b></p> <ul style="list-style-type: none"> <li>• Have acquired learning capabilities that enable them to carry out project activities in companies, public institutions or in distributed development communities;</li> </ul>
--------------------------	---

<b>ASSESSMENT</b>	<p>The assessment is based on lab assessment (50%) and the written exam (50%).          The lab assessment consists of bi-weekly assignments and a project. The project addresses the advanced techniques reviewed in the course.          The written exam assessment evaluates the ability of students to understand and use the techniques reviewed in the course.          To access the final written exam, the students must have passed (15 points or more) the lab assessment.</p>
-------------------	--

	<p>In case the lab assessment is positive but the final written exam is not positive, the lab grade is valid for all three regular exam sessions.</p> <p>There is a mid-term exam. The mid-term exam accounts for 50% of the final written exam. In case the mid-term exam is positive (15 points or more), the grade is valid for all three regular exam sessions.</p>
<b>ASSESSMENT LANGUAGE</b>	English
<b>EVALUATION CRITERIA AND CRITERIA FOR AWARDING MARKS</b>	<p>The final mark is composed of:</p> <ul style="list-style-type: none"> <li>- 50% lab assessment</li> <li>- 50% written exam</li> </ul> <p>In order to be admitted to the written exam, the student must obtain at least 50% of the lab assessment (50% of the lab assessment correspond to 25% of the total assessment).</p> <p>The lab assessment is based on the assessment of the project. In case the student receives less than 100% from the project assessment, additional points can be awarded based on the results of the bi-weekly lab assignments, but not more than 100% (which corresponds to 50% of the total assessment).</p> <p>The student can take the written exam at the end or take also the mid-term exam during the semester, for which up to 50% of the written exam points can be awarded.</p> <p>The lab assessment evaluates the ability of students to apply methods of team working, to integrate and use technologies to professionally develop programs, and to apply advanced techniques of programming.</p> <p>The written assessment evaluates the ability of students to understand and use the different language techniques needed to develop a project for the requirements to code. Students are evaluated by their understanding of advanced constructs in java.</p>

<b>REQUIRED READINGS</b>	Lecture notes will be handed out during the course.
<b>SUPPLEMENTARY READINGS</b>	<p>Paul Deitel; Harvey Deitel, Java™ How To Program (Early Objects), Tenth Edition, Prentice Hall, 2014, ISBN 978-0-13-381303-6, available through Library access to Safari Books Online, Permanent link:  <a href="http://ubz-primohosted.exlibrisgroup.com/UNIBZ:All:39UBZ_ALMA_DS51129685850001241">http://ubz-primohosted.exlibrisgroup.com/UNIBZ:All:39UBZ_ALMA_DS51129685850001241</a></p>
<b>SOFTWARE USED</b>	<ul style="list-style-type: none"> <li>- Eclipse IDE</li> <li>- JDK</li> </ul>