unibz

Fakultät für Informatik
Facoltà di Scienze e Tecnologie informatiche
Faculty of Computer Science

# SYLLABUS
# COURSE DESCRIPTION

| | |
|---|---|
| **COURSE TITLE** | Advanced Programming |
| **COURSE CODE** | 75035 |
| **SCIENTIFIC SECTOR** | INF/01 |
| **DEGREE** | Bachelor in Computer Science and Engineering |
| **SEMESTER** | 2nd Semester |
| **YEAR** | 1st |
| **CREDITS** | 8 |

| | |
|---|---|
| **TOTAL LECTURING HOURS** | 48 |
| **TOTAL LAB HOURS** | 24 |
| **PREREQUISITES** | Students are familiar with basic knowledge of object oriented programming and Java.<br>This material is taught in the following undergraduate courses:<br>Introduction to Programming |
| **COURSE PAGE** | [https://ole.unibz.it/](https://ole.unibz.it/) |

| | |
|---|---|
| **SPECIFIC EDUCATIONAL OBJECTIVES** | • Type of course: “di base” for L-31 and L-08<br>• Scientific area: “formazione informatica di base” for L-31 and “matematica, informatica e statistica” for L-8<br>This course 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) of the software developed. |

| | |
|---|---|
| **LECTURER** | Marko Tkalcic, office POS 1.13<br>marko.tkalcic@unibz.it<br>http://www.inf.unibz.it/~tkalcic/ |
| **SCIENTIFIC SECTOR OF THE LECTURER** | INF/01 |
| **TEACHING LANGUAGE** | English |

| | |
|---|---|
| **OFFICE HOURS** | Tuesday, 16:00 |
| **TEACHING ASSISTANT** | Marko Tkalcic, marko.tkalcic@unibz.it<br>Markus Gritsch, markusjohann.gritsch@unibz.it |
| **OFFICE HOURS** | Marko Tkalcic: Tuesday, 16:00, POS 1.13, Piazza Domenicani 3;<br>Markus Gritsch: Friday, 16:00 – 17:00, POS 1.04, Piazza Domenicani 3; |
| **LIST OF TOPICS COVERED** | - Object Orientation principles in software life cycle<br>- Memory models in Java<br>- Virtual functions, late binding, overriding, and overloading<br>- Exception handling<br>- Use of API<br>- Reflection and runtime type identification<br>- Coding with design patterns<br>- Generics and collections |
| **TEACHING FORMAT** | Frontal lectures and projects in teams |

| | |
|---|---|
| **LEARNING OUTCOMES** | **Knowledge and understanding**<br>- Have a solid knowledge of the most important data structures and programming techniques;<br>- Know the fundamental principles of programming;<br>**Applying knowledge and understanding**<br>- Be able to develop small and medium size programs<br>- Be able to solve problems using programming;<br>- Be able to conduct simple experiments about information systems by collecting measures about the behaviour of the system;<br>**Making judgments**<br>- Be able to work autonomously according to the own level of knowledge<br>- Be able to take the responsibility for software development projects<br>**Communication skills**<br>- Be able to work in teams to implement software systems;<br>- Be able to use modern communication systems.<br>**Learning skills**<br>- Have acquired learning capabilities that enable them to carry out project activities in companies, public institutions or in distributed development communities |

| | |
|---|---|
| **ASSESSMENT** | The assessment is based on lab assessment (50%) and the written exam (50%), with an optional mid-term written exam.<br><br>The lab assessment consists of bi-weekly assignments and a project. The project addresses the advanced techniques reviewed in the course. 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.<br><br>The written exam assessment evaluates the ability of students to understand and use the techniques reviewed in the course. Students are evaluated by their understanding of advanced constructs in java. |

| | |
|---|---|
| | 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 (18 points or more), the grade is valid for all three regular exam sessions. |
| **ASSESSMENT LANGUAGE** | English |
| **EVALUATION CRITERIA AND CRITERIA FOR AWARDING MARKS** | The final mark is composed of:<br>- 50% lab assessment<br>- 25% mid-term exam (optional)<br>- 25% written exam (or 50% in case of failure in the Mid-Term exam)<br>To be admitted at the final written exam, the students must have passed (18 points or more) the lab assessment. 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.<br>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. |

| | |
|---|---|
| **REQUIRED READINGS** | Lecture notes will be handed out during the course.<br>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: http://ubz-primo.hosted.exlibrisgroup.com/UNIBZ:All:39UBZ_ALMA_DS51129685850001241 |
| **SUPPLEMENTARY READINGS** | Joshua Bloch, Effective Java (2nd Edition), Addison Wesley, 2008 |
| **SOFTWARE USED** | - Eclipse IDE<br>- JDK |