# Syllabus

## *Kursbeschreibung*

| | |
|---|---|
| **Titel der Lehrveranstaltung** | Object Oriented and Functional Programming |
| **Code der Lehrveranstaltung** | 76277 |
| **Zusätzlicher Titel der Lehrveranstaltung** | |
| **Wissenschaftlich-disziplinärer Bereich** | INF/01 |
| **Sprache** | Englisch |
| **Studiengang** | Bachelor in Informatik |
| **Andere Studiengänge (gem. Lehrveranstaltung)** | |
| **Dozenten/Dozentinnen** | dr. Julien Louis Michel Corman, JulienLouisMichel.Corman@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/37010 |
| **Wissensch. Mitarbeiter/Mitarbeiterin** | |
| **Semester** | Zweites Semester |
| **Studienjahr/e** | 1 |
| **KP** | 6 |
| **Vorlesungsstunden** | 40 |
| **Laboratoriumsstunden** | 20 |
| **Stunden für individuelles Studium** | 90 |
| **Vorgesehene Sprechzeiten** | |
| **Inhaltsangabe** | This course belongs to the type "Attività formativa caratterizzante" and the subject area is "Scientifico-Tecnologico".<br><br>The course is designed to help students develop generic, object-oriented and functional programming skills. After completing the course, students should be able to implement algorithms to solve simple programming problems and select appropriate data |

| | |
|---|---|
| | structures, write readable, concise, modular, and documented code. |
| **Themen der Lehrveranstaltung** | • Object-oriented design: objects, classes, interfaces, inheritance and polymorphism<br>• Abstract data types (set, list, associative array, queue, ...) and related data structures (linked list, hash table, ...),<br>• Composite types, type inference, generics<br>• Basic input/output, serialization, streams, error handling, custom exceptions, debugging<br>• Introduction to functional programming: function composition, recursion, currying, closures, function types<br>• Functional principles applied to object-oriented programming: immutability, pure functions, lambda abstractions |
| **Stichwörter** | object-oriented programming, functional programming |
| **Empfohlene Voraussetzungen** | The course requires knowledge of the basics of imperative and object-oriented programming, particularly in Java as taught in the "Introduction to Programming" course. |
| **Propädeutische Lehrveranstaltungen** | |
| **Unterrichtsform** | The course includes frontal lectures and lab exercises. |
| **Anwesenheitspflicht** | Attendance to lectures and labs is optional. However, non-attending students should contact the lecturer at the start of the course to discuss the modality of their independent study. The evaluation process is slightly different for attending and non-attending students. It is described in the fields "Assessment" and "Evaluation criteria and criteria for awarding marks" below. |
| **Spezifische Bildungsziele und erwartete Lernergebnisse** | Knowledge and Understanding<br>- D1.2: Know in details the fundamental principles of programming<br><br>- D1.3: Have a solid knowledge of the most important data structures and programming techniques<br><br>Applying knowledge and understanding<br>- D2.2: Be able to develop small and medium size programs using different programming languages and paradigms.<br>- D2.3: Be able to solve problems using programming methodologies. |

| | |
|---|---|
| | Ability to make judgments<br>- D3.1: Be able to collect and interpret useful data and to judge information systems and their applicability.<br>- D3.2: Be able to work autonomously according to the own level of knowledge and understanding.<br>- D3.3: Be able to take the responsibility for development of projects or IT consulting.<br><br>Communication skills<br>- D4.1: Be able to use one of the three languages English, Italian and German, and be able to use technical terms and communication appropriately.<br>- D4.4: Be able to work in teams for the realization of IT systems.<br><br>Learning skills<br>- D5.1: Have developed learning capabilities to pursue further studies with a high degree of autonomy.<br>- D5.2: Have acquired learning capabilities that enable to carry out project activities in companies, public institutions or in distributed development communities.<br>- D5.3: Be able to follow the fast technological evolution and to learn cutting edge IT technologies and innovative aspects of last generation information systems. |
| **Spezifisches Bildungsziel und erwartete Lernergebnisse (zusätzliche Informationen)** | |
| **Art der Prüfung** | The assessment is based on assignments, which focus on topics taught during lectures and are designed to motivate students to study throughout the semester while consolidating the theoretical concepts covered in class. The assignments are individual. Additionally, a written evaluates whether students have acquired the expected notions and skills. |
| **Bewertungskriterien** | Final marks will be calculated as follows: up to 60 points will be awarded for assignments, and up to 40 points for the written exam.<br>Students who attend the course and labs will benefit from an easier grading scheme, but may be asked in return to explain the code that they submitted for some assignments (during the labs). |

| | |
|---|---|
| **Pflichtliteratur** | • Lecture material on the course's website<br>• Kathy Sierra, Bert Bates, and Trisha Gee. Head First Java: A Brain-Friendly Guide. O'Reilly Media, Sebastopol, CA, 3rd edition, June 2022. ISBN 978-1-4919-1077-1.<br>• Herbert Schildt. Java: The Complete Reference. McGraw Hill, 11th edition, 2018. |
| **Weiterführende Literatur** | Joshua Bloch. Effective Java. Addison-Wesley Professional, Boston, 3rd edition, 2017. ISBN 978-0-13-468599-1. |
| **Weitere Informationen** | - IntelliJ IDEA (https://www.jetbrains.com/idea/)<br>- Visual Studio Code (https://code.visualstudio.com)<br>- JDK 21 (https://openjdk.org/projects/jdk/17/)<br>- Maven (https://maven.apache.org)<br>- As operating system, Linux or MacOS are recommended |
| **Ziele für nachhaltige Entwicklung (SDGs)** | Hochwertige Bildung |