COURSE PRESENTATION FORM

COURSE NAME  
Introduction to Programming

COURSE CODE  
70132 / 70003

LECTURER  
Diego Calvanese

TEACHING ASSISTANT  
Christian Mair, Davide Martinenghi

TEACHING LANGUAGE  
English

CREDIT POINTS  
8 (BSc new study plan) / 6 (BSc old study plan)

LECTURE HOURS  
48

EXERCISE HOURS  
24

PREREQUISITES  
There are no specific prerequisites. Basic notions of mathematics and set theory will be used.

OBJECTIVES  
The objective of the Introduction to Programming course is to teach the fundamental principles of programming, making use of the typical aspects of the object-oriented, the functional, and the imperative programming paradigms. Such basic principles are presented by referring to the Java programming language.

SYLLABUS  
For students taking the 6CFU exam (old study plan), the syllabus covers the following topics: introduction to programming and to Java; use of objects; definition of methods and classes; primitive data types; conditional statements; loop statements; arrays; files and input/output; program errors and exceptions.

For students taking the 8CFU exam (new study plan), the syllabus additionally covers the topics: recursion; dynamic arrays and linked lists.

TEACHING FORMAT  
Frontal lectures; exercises in the computer laboratory

ASSESSMENT  
Final examination consisting of two parts: a lab examination (pass/fail) and a written examination (100%). The two parts have to be taken at the same exam session. The lab examination is also offered as a midterm examination and in this case, if passed, counts for all three exam sessions of the academic year.

READING LIST  

Suggested Readings: An additional book covering in depth all aspects of the Java language is suggested, such as Absolute Java. Walter Savitch. Addison Wesley, 2nd Edition 2005.

SOFTWARE USED  
• Java 2 Standard Edition 5.0 SDK
• BlueJ development environment

LEARNING OUTCOME  
After the course, students will know the fundamental principles of object-oriented programming, including the use of control structures, functional abstraction, classes and methods, and basic data structures, and will be able to put them into practice, by writing programs in Java.