Academic subject: Computer Science			
Degree Class: L-35 – Scienze Matematiche		**Degree Course:** Mathematics	**Academic Year:** 2020/2021
		Kind of class: Mandatory	**Year:** 1 / **Period:** 1
			ECTS: 6 divided into **ECTS lessons:** 4 **ECTS exe/lab/tutor:** 2
Time management, hours, in–class study hours, out–of–class study hours lesson: 32 exe/lab/tutor: 16 in–class study: 48 out–of–class study: 102			
Language: Italian	**Compulsory Attendance:** no		
Subject Teacher: Angelo Cardellicchio	**Tel:** +39 080 596 3312 **e–mail:** angelo.cardellicchio@uniba.it	**Office:** Polytechnic University of Bari, DEI, Ex-Architettura, Room 4.12	**Office days and hours:** Monday – Wednesday 16:30 – 17:30 By appointment
Prerequisites: Basic mathematics notions provided by Italian high schools			
Educational objectives: Learn basic knowledge of computers architecture and networks. Learn basic concepts concerning programming methods, techniques. Ability to apply such concepts for solving problems by means of C/C++ and Python languages. Basic notions and concepts about computability and algorithms.			
Expected learning outcomes (according to Dublin Descriptors)	**Knowledge and understanding:** Learning basic concepts concerning programming methods and techniques. **Applying knowledge and understanding:** Ability to apply learned concepts for solving and implementing problems solutions. **Making judgements:** Capability to judge the consistency of the logical structure used in building algorithms. Capability to identify the proper conceptual tools for solving programming problems. **Communication:** Learning the specific computer science language and methods for understanding textbooks, for explaining the learned knowledge, for describing, analyzing and solving algorithmic problems. **Lifelong learning skills:** Acquiring suitable learning methods, supported by text consultation and by solving the exercises and questions periodically suggested during the course.		
Course program Introduction to the course Introduction to computers architecture and networks Introduction to programming Flow charts Structured programming Programming constructs Data structures Variables and functions Introduction to computability Computational complexity			

Turing Machine
Recursive functions

Introduction to C

Introduction to C++

Introduction to Python

Teaching methods:
Lectures and exercise sessions

Auxiliary teaching:
Didactic material available at the following URL: https://anhelus.github.io/informatica-dm-uniba/

Assessment methods:
Final year project.
Exams with (possible) written exercises and oral discussion.

Bibliography:
Andrew J. Tanenbaum, Architettura dei calcolatori. Un approccio strutturale.
Andrew J. Tanenbaum, Reti di calcolatori.
Deitel, Deitel, C++. Fondamenti di programmazione.

Slides and other material cited / deployed during lectures.