COURSE DESCRIPTION – ACADEMIC YEAR 2023/2024

Course title	**Fundamentals of Programming I**
Course code	42405
Scientific sector	INF/01
Degree	Bachelor in Electronics and Cyberphysical Systems
Semester	1
Year	1
Credits	6
Modular	Yes

Total lecturing hours	40
Total lab hours	20
Attendance	Highly recommended, e.g., for using the electronic material of the course.
Prerequisites	There are no specific prerequisites. Basic notions of mathematics and set theory will be used.
Course page	Teams

Specific educational objectives	Type: "attività formativa di base" Scientific area: "Matematica, informatica e statistica" The course is designed for acquiring professional skills and knowledge. The objective of the course is to teach the fundamental principles of programming, with a focus on structural programming, and tools to support the development of software in Python. Students will learn how to solve computational problems with well-designed programs. Learning is based on practice with progressive exercises, from very simple ones to more complex. Exercises guide students towards the prototyping of simple software solutions for basic cyber-physical systems that perceive data via physical devices (e.g., temperature sensor, humidity sensor), processes them, and reacts via physical devices (e.g., LEDs, buzzers). The final objective for the student is to acquire the ability to translate a set of requirements into a software solution for basic cyber-physical systems.

Lecturer	Rosella Gennari
Contact	gennari@inf.unibz.it
Scientific sector of lecturer	INF 01 (by contract), ING-INF 05 (I have also got this habilitation as associate professor, besides that in INF 01)
Teaching language	
Office hours	To arrange beforehand during class hours or by email.
Lecturing assistant (if any)	None
Contact LA	None
Office hours LA	None

List of topics	1. Introduction to: hardware and software, with computer organisation; data hierarchy; machine languages, assembly languages, high-level programming languages. 2. Introduction to Python: interactive mode, script mode, Jupyter. 3. Introduction to different programming paradigms, focusing on the structured programming paradigm. 4. Structured programming: basic data types, variables, constants, operators and expressions; standard input/output handling; control flow structures; file and error handling. 5. Basic data structures/types of Python: (1) lists, (2) dictionaries, (3) tuples, (4) sets.Subroutines and functions in Python (with/without parameters; with/without return); functions and basic recursion in Python, e.g., some combinatorics. 6. Basics of computational thinking to solve a computational problem and program a resolution in Python and Python-based languages. The above will be delivered meanwhile acquiring knowledge of the working of a physical-computing board and how to program it in a Python-based language (MicroPython or CircuitPython), to perceive data via physical input devices (e.g., temperature sensor, humidity sensor), process data and react via physical output devices (e.g., LEDs, buzzers) and, possibly, how to plot data in relation to the types of sensed data.
Teaching format	Frontal lectures, labs with exercises for projects.

Learning outcomes	**Knowledge and understanding** • Know the fundamental principles of programming. • Know different programming paradigms and models of computation. • Have a solid knowledge of the most important data structures and programming techniques. **Applying knowledge and understanding** • Be able to solve problems using programming. • Be able to develop small and medium size programs starting from given requirements. **Making judgements** • Be able to collect and interpret useful data and to judge information systems and their applicability. • Be able to identify an appropriate programming paradigm and data structures to solve a given problem. **Communication skills** • Be able to describe and motivate the software design choices. • Be able to properly document a software artifact to ensure its integration in more complex systems. **Learning skills** • Be able to learn how to use different procedural programming languages in autonomy, by identifying and understanding the relevant literature.

Assessment	A programming project and a final written exam for Fundamentals of Programming 1 and 2. First-year students can take an intermediate exam, consisting of the written part of Fundamentals of Programming 1. The project will assess the learning outcomes related to the application of the acquired knowledge, the ability to make judgments and the communication and learning skills. The written exam contains verification questions, transfer of knowledge questions and exercises. The written examination will assess the learning outcomes related to knowledge and understanding, applying knowledge, and understanding, and those related to the student's ability to learn.
Assessment language	English
Assessment Typology	Collegial
Evaluation criteria and criteria for awarding marks	Project counts for 50% of mark, and the final exam (written) for 50% of the mark. In case of a positive mark the project will count for three exam sessions. A project is evaluated in term of quality of the solution: easy to use, meaningfulness of the implemented functions, quality of the code (according to the principles that will be illustrated during the lectures). Written exam questions will be evaluated in term of correctness and clarity.

Required readings	Material is provided during the course. Subject Librarian: David Gebhardi, David.Gebhardi@unibz.it and Ilaria Miceli, Ilaria.Miceli@unibz.it
Supplementary readings	Additional material will be provided during the course