COURSE DESCRIPTION – ACADEMIC YEAR 2024/2025

Course title	**Laboratory of Fundamentals of Programming**
Course code	42612
Scientific sector	INF/01
Degree	Bachelor in Wood Engineering
Semester	2
Year	1
Credits	3
Modular	No

Total lecturing hours	30 (see related syllabus)
Total lab hours	30
Attendance	Not compulsory but strongly advisable.
Prerequisites	Basics of mathematics.
Course page	Microsoft Teams, code communicated at the start of the course.

Specific educational objectives	The course belongs to the type "caratterizzanti – discipline informatiche". By following the latest European Commission and national recommendations and guidelines on computational thinking and computing education, the course gives a general overview of scientific contents and computing technologies, which are relevant for tomorrow's citizens. The overall goal of the course is to **empower different students** to tackle a simple computational problem and develop a solution for it, critically and collaboratively. The specific objectives to achieve the goal are as follows. 1) First, the course aims to provide participants with **basic knowledge of computing** to **understand** a basic computational problem, that is, to analyse it and abstract away what needed for developing a basic computing solution for it. 2) Second, the course aims to enable students to **develop basic computing solutions** for different problems, which requires them to specify and program them. Third the course aims to enable students to **collaborate** in the analysis of problems and development of solutions, and to **critically** reflect on what they are doing.

Lecturer	Rosella Gennari
Contact	B1.3.21, gennari@inf.unibz.it
Scientific sector of lecturer	
Teaching language	English
Office hours	After each lecture, by prior appointment
Lecturing Assistant (if any)	
Contact LA	
Office hours LA	

List of topics	1. Introduction to a resource-constrained implementation of the Python programming language targeting embedded systems, specifically programmable microcontrollers such as the micro:bit. 2. Fundamental input/output (I/O) operations for microcontroller peripherals, including interfacing with sensors (e.g., motion sensors) and actuators (e.g., LED strips). 3. Core control flow constructs, data structures, and subroutine mechanisms for microcontroller programming. The above is tackled for covering the basics of computing to critically understand a computational problem and develop a resolution in a Python-based programming language for programmable microcontrollers.
Teaching format	In-presence, lecture and workshop-based.

Learning outcomes	**Intended Learning Outcomes (ILO)** **Applying knowledge and understanding** **1.** Be able to analyse basic computational problems. **2.** Be able to specify one among many computational solutions. **3.** Be able to program computational solutions. **4.** Be able to understand computational solutions. **Making judgements** **5.** Be able to collect and interpret useful data and to judge computational solutions and their applicability. **6.** Be able to identify critical aspects in the development process and take a critical stance towards what is developed. **Communication skills** **7.** Be able to describe and motivate choices. **8.** Be able to properly document a computing solution.

Assessment	**Formative assessment** Not foreseen **Summative assessment** Assessment format: project, related to all the listed topics (1–3). The project is different for attending students and non-attending students. A student is attending if the student attends >70% lab hours, and the student actively tackles project-related tasks during lab hours. Attending students: 1. The project utilises course materials provided during lab sessions, specifically, microcontrollers, their integrated I/O peripherals, further external I/O devices. 2. Project development and documentation are primarily conducted during scheduled lab sessions. 3. Project discussion is conducted orally, during the examination.

	Non-attending students: 1. The project utilises a programmable microcontroller running a Python-based language, that students need to acquire on their own, that is, the latest micro:bit version (at least V2.2). 2. Project development and documentation are uploaded in the course repository before the exam, following the specifications of the course teacher. 3. Project discussion is conducted orally, during the examination. In case of a pass mark, the outcome will count for all 3 regular exam sessions. The total number of hours the student devotes to the course is #CFU * 25 (e.g., 150 hours for a 6 CFU course), including: a. the time spent in class; b. the preparation of the project; c. the time for independent study.
Assessment language	English
Assessment Typology	Monocratic
Evaluation criteria and criteria for awarding marks	The outcome is pass or fail (no marks). The following **ILO**s will be considered criteria for evaluating the outcome, all with a similar weight: 1. Applying knowledge and understanding of computing in the project 2. Making judgements, by taking a critical reflective stance on computing in the project 3. Communication skills and master of the technical language in the project 4. Learning skills in approaching one's project and deepening one's knowledge in autonomy in developing the project

Required readings	Material provided by the lecturer.
Supplementary readings	Online resources suggested by the lecturer, e.g., https://microbit-challenges.readthedocs.io/en/latest/introduction/letter.html
Software used	MicroPython or CircuitPython, basic IDEs.