SYLLABUS
COURSE DESCRIPTION YEAR 2023/2024

COURSE TITLE	**Maker Lab**
COURSE CODE	76244
SCIENTIFIC SECTOR	ING-INF/01
DEGREE	Bachelor in Computer Science
SEMESTER	1st
YEAR	2nd
CREDITS	3

TOTAL LECTURING HOURS	30
TOTAL LAB HOURS	-
ATTENDANCE	Generally, attendance is not compulsory, but highly recommended also to exploit the physical computing devices part of the course, e.g., RPi computers or microcontrollers which cannot be brought home. Non-attending students have to contact the lecturer at the start of the course to agree on the modalities of the independent study; they will have to buy the course material on their own.
PREREQUISITES	Basic knowledge of programming languages and computer systems architectures
COURSE PAGE	https://ole.unibz.it/ http://www.inf.unibz.it/~gennari/makerlab.html

SPECIFIC EDUCATIONAL OBJECTIVES	Type of course: "Ulteriori attività formative" for L-31 Scientific area: "conoscenze utili per l'inserimento nel mondo del lavoro" for L-31 The course is designed for acquiring professional skills and knowledge. It gives general practical knowledge and skills necessary for developing basic smart objects for IoT.

LECTURER	Soufiane Krik
SCIENTIFIC SECTOR OF THE LECTURER	INF/01
TEACHING LANGUAGE	**Italian**

OFFICE HOURS	Office hours take place upon appointment
TEACHING ASSISTANT	Max Dorfmann
OFFICE HOURS	
LIST OF TOPICS COVERED	• Basics of programming for physical computing and interactions with the world using Python 3 • Basics of electronics for physical computing: interruptors, sensors, actuators and hats • Basics of interaction design and development for physical computing
TEACHING FORMAT	The teaching format is made up of frontal lectures and workshops. The course adopts a learning-by-doing approach with in-presence formative feedback. In practice, during classes, students can tackle numerous scaffolding challenges so as to learn by doing. In order to tackle them, students must be equipped with their own computer. Physical computing devices are provided to attending students. The formative feedback on resolutions is given by the course instructors during class hours.

LEARNING OUTCOMES	**Knowledge and understanding**: • Know the main methods for the design of interactive smart objects for IoT. **Applying knowledge and understanding** • Be able to apply the own knowledge in different working contexts; • Be able to coordinate small project teams and to interact with members of the group; • Be able to apply interactive design principles and patterns for IoT solutions and smart objects. **Making judgments** • Be able to work autonomously according to the own level of knowledge and understanding. **Communication skills** • Be able to use one of the three languages English, Italian and German, and be able to use technical terms and communication appropriately; • Be able to work in teams for the realization of IT systems. **Learning skills** • Have acquired learning capabilities that enable to carry out project activities in companies, public institutions or in distributed development communities.

ASSESSMENT	The assessment is related to the following: (1) knowledge of Python; (2) the ability to write a Python program for a given problem; (3) the ability to correct a Python program for a given problem; (4) the ability to optimise a Python program for a given problem; (5) knowledge of RPi and hats; (6) the ability to understand the design of a smart object with RPi and Python; (7) the ability to evaluate the design of a smart object with RPi and Python; (8) the ability to fix the design of a smart object with RPi and Python; (9) the ability to develop a smart object with RPi and Python; (10) the ability to interface other physical computing devices such as Arduino UNO with RPi and Python Attending students (for more than 80% of the course) solve exercises during the course. By using their resolutions, they work on a programming project which they need to present and discuss. Non-attending students solve a written exam with several exercises.
ASSESSMENT LANGUAGE	**Italian**
EVALUATION CRITERIA AND CRITERIA FOR AWARDING MARKS	Evaluation criteria for the quality of resolutions will be: clarity, correctness and completeness of programs, besides usability and elaboration requirements of their smart objects. The quality of the resolutions and the ability of the student to discuss them will determine whether the student passes or fails the course.

REQUIRED READINGS	Downey, Think Python, 2nd Edition. Available via the Safari library. Shaw. Learn Python the Hard Way. Available via the Safari library.
SUPPLEMENTARY READINGS	
SOFTWARE USED	SOFTWARE: Python, C++. HARDWARE: Raspberry Pi (RPi) and Arduino UNO kits, available to groups of attending students during class hours only.