Universitat de Lleida

DEGREE CURRICULUM

# PROGRAMMING AND COMMUNICATIONS I

Coordination: BEJAR TORRES, RAMON

Academic year 2021-22

# PROGRAMMING AND COMMUNICATIONS I 2021-22

## Subject's general information

| | |
|---|---|
| **Subject name** | PROGRAMMING AND COMMUNICATIONS I |
| **Code** | 102133 |
| **Semester** | 1st Q(SEMESTER) CONTINUED EVALUATION |
| **Typology** | |

| Degree | Course | Character | Modality |
|---|---|---|---|
| Bachelor's Degree in Automation and Industrial Electronic Engineering | 4 | OPTIONAL | Attendance-based |

| | |
|---|---|
| **Course number of credits (ECTS)** | 6 |
| **Type of activity, credits, and groups** | |

| Activity type | PRAULA | TEORIA |
|---|---|---|
| Number of credits | 3 | 3 |
| Number of groups | 1 | 1 |

| | |
|---|---|
| **Coordination** | BEJAR TORRES, RAMON |
| **Department** | COMPUTER SCIENCE AND INDUSTRIAL ENGINEERING |
| **Teaching load distribution between lectures and independent student work** | 6 ECTS = 25x6 = 150<br>- 60 hours of on-class activities<br>- 90 hours of autonomous activities |
| **Important information on data processing** | Consult this link for more information. |
| **Language** | English |

| Teaching staff | E-mail addresses | Credits taught by teacher | Office and hour of attention |
|---|---|---|---|
| BEJAR TORRES, RAMON | ramon.bejar@udl.cat | 3 | |
| ORELLANA TRULLOLS, GUILLEM | got2@alumnes.udl.cat | 3 | |

## Subject's extra information

**Previous Requirements**
The student must have taken and passed the subjects from the "basic training" module and the industrial computing subject of the "common training" module.

For students who come from other university degrees, you must have completed subjects covering basic knowledge about fundamentals of computer programming, have basic knowledge about the Linux operating system and electronic circuits and digital sensors.

## Learning objectives

**Learning Goals**

**Expected learning outcomes related to the strategic transversal competences:**

- The student is able to learn and work with technical documentation in English on programming languages on Linux environment (Competences UdL2, EPS4).

- The student is able to work in teams to carry out the development of a computer system composed of different subsystems in which there to apply knowledge from different fields (Competence EPS9).

- The student is able to prepare presentations in English to show aspects main programs developed for other equipment engineers can understand their solutions. (Competences UdL2, UdL3).

- The student is able to generalize basic algorithmic schemes to apply them in different contexts and problems from the ones initially seen (Competence EPS4).

**Expected learning outcomes linked to specific competencies:**

- The student is able to understand the basic features of the functioning of operating systems multitasking, multiuser based on the Linux kernel (Competences GEEIA3 and GEEIA34).

- The student is able to integrate the knowledge of circuits, sensors and processes industrial on knowledge of computer programming for address the full development of small automated systems monitoring / control software based processes on Raspberry Pi computers or microcomputers (Competences GEEIA3, GEEIA28 and GEEIA34).

## Competences

**Strategic competences UdL:**

- UdL2. Knowledge of a foreign language.
- UdL3. Knowledge of ICT.

**Transversal competences EPS:**

- EPS4. Have the learning abilities needed to start superior studies or improve the academic learning with a certain autonomous degree.
- EPS9. Ability to work in teams, both as a unidisciplinary and multidisciplinary.

**Specific competences GEEIA:**

- GEEIA3 Basic knowledge on using and programming computers, operating systems, databases and software with applications in engineering.
- GEEIA28. Applied knowledge of industrial computing and communications.
- GEEIA-EPS34. Knowledge of the fundamentals of computer systems and applications.

## Subject contents

- Introduction
- Variables, expressions and statements
- Conditional execution
- Functions
- Loops and Iterations
- Strings
- Files
- Lists, Dictionaries and Tuples
- Filtering Information
- RaspberryPI setup and configuration
- Getting data from sensors with the RaspberryPI GPIO

## Methodology

Face-to-face activities:

- Problems
- Laboratories
- Tests and evaluation

On-line activities:

- Master classes with videos on the virtual campus.
- Session of doubts / resolution of problems with the videoconferencing tool of the virtual campus.

Autonomous work:

- Compulsory programming projects
- Small programming exercises

## Development plan

Weeks 1-4: Introduction to python programming: basic elements
Weeks 5-8: Functions, lists, dictionaries and files
Week 9: Oral presentation of the first project
Weeks 10-11: Object-oriented programming with python
Weeks 12-15: Programming with digital sensors with the Raspberry Pi
Week 16: Development of the second project in the Raspberry Pi
Week 17: Oral presentation of the second project

## Evaluation

Evaluation activities

| Acr. | Evaluation activity | Weight | Minimum grade | In group | Mandatory |
|---|---|---|---|---|---|
| *P1* | *Programming assignment* | (27+13)% | NO | YES (1) | YES |
| *P2* | *Programming assignment* | (27+13)% | NO | YES (1) | YES |
| *OR1* | *Oral Presentation and questions* | 10% | NO | YES (2) | YES |
| *OR2* | *Oral Presentation and questions* | 10% | NO | YES (2) | YES |

**Final Grade** = 0,4*P1 + 0,4*P2 + 0,1*OR1 + 0,1*OR2
(1) : In each programming assignment, each member will have to answer individually some questions, that will give 1/3 of the points of the programming assignment. That is, the 40% weight in the final grade, is composed of a 27 % weight for the code of the program (same grade for both members) and 13% for the answers to the questions about the program (individual grades for each member).
(2) : Each member of the group will perform a different part of the presentation, and individual questions will have to be answered by both members.

## Bibliography

Basic Bibliography:


- Mark Lutz. Learning Python 4th Edition. OReilly - 2009.
- Raspberry Pi Cookbook - Simon Monk - O'Reilly- 2014


Some free on-line books for learning python:


- Dive into python. http://www.diveintopython.net/
- A Byte of Python - Una mica de Python. http://moiatgit.github.io/byte_of_python_120.cat/


On-line resources.

- Python: http://docs.python.org/2.7/
- Raspberry Pi: http://www.raspberrypi.org/

The free on-line resources are enough to follow this subject, but if you want to have a good book for developing and understanding many classes of problems and programs on the RsPI, choose the *Raspberry Pi Cookbook*