Universitat de Lleida

DEGREE CURRICULUM

PROGRAMMING AND COMMUNICATIONS I

Coordination: BEJAR TORRES, RAMON

Academic year 2017-18

Subject's general information

Subject name	PROGRAMMING AND COMMUNICATIONS I
Code	102133
Semester	1st Q(SEMESTER) CONTINUED EVALUATION
Typology	<table><tr><th>Degree</th><th>Course</th><th>Typology</th><th>Modality</th></tr><tr><td>Bachelor's Degree in Automation and Industrial Electronic Engineering</td><td>4</td><td>OPTIONAL</td><td>Attendance-based</td></tr></table>
ECTS credits	6
Groups	1GG
Theoretical credits	3
Practical credits	3
Coordination	BEJAR TORRES, RAMON
Department	INFORMATICA I ENGINYERIA INDUSTRIAL
Teaching load distribution between lectures and independent student work	6 ECTS = 25x6 = 150 - 60 hours of on-class activities - 90 hours of autonomous activities
Important information on data processing	Consult this link for more information.
Language	English
Office and hour of attention	To arrange with the teacher

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
BEJAR TORRES, RAMON	ramon@diei.udl.cat	7,2	office 3.23 appointments by email

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 an 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

Learning activities
Face to face activities (40%): The percentages associated to each one of the activities are computed over 100%

- Master class (42,5%)
- Problems (25%)
- Laboratory (25%)
- Tests and evaluation (7,5%)

Autonomous work (60%): The percentages associated to each one of the activities are computed over 100%

- Working on the mandatory programming assignments (80%)
- Solving small programming problems (20%)

Development plan

Week	Description	Face-to-Face Activity	Autonomous Activity	Hours (F and A)
1	Presentation and introduction to programming microcomputers	Lectures and programming laboratory	**Solve Exercises**	**4 2**
2	Python expressions	Lectures and programming laboratory	**Solve Exercises**	4 6
3	Python conditional expressions	Lectures and programming laboratory	**Solve Exercises**	4 6
4	Python Functions	Lectures and programming laboratory	**Solve Exercises**	4 6

5	Python loops	Lectures and programming laboratory	**Solve Exercises**	**4 6**
6	Python strings	Lectures and programming laboratory	**Solve Exercises**	4 6
7	Python lists	Lectures and programming laboratory	Work on 1st programming assignment **Solve Exercises**	4 8
8	Python files	Lectures and programming laboratory	Work on 1st programming assignment **Solve Exercises**	4 8
9		**Oral Presentation of programming assignment**	Work on 1st programming assignment - work on presentation	4 8
10	Python dictionaries and tuples	Lectures and programming laboratory	**Solve Exercises**	4 6
11	OOP Python programming	Lectures and programming laboratory	**Solve Exercises**	4 6
12	RSPI and sensors	Lectures and programming laboratory	**Solve Exercises**	4 6
13	RSPI and sensors	Lectures and programming laboratory	**Solve Exercises** **Work on 2nd programming assignment**	4 10
14	RSPI and sensors	Lectures and programming laboratory	**Work on 2nd programming assignment**	4 8
15	RSPI and sensors	Lectures Lab3:	**Work on 2nd programming assignment**	4 8
16			**Work on 2nd programming assignment**	- 6
17		**Oral Presentation of programming assignment**	work on presentation	2 4
18				
19				

Evaluation

Evaluation activities

Acr.	Evaluation activity	Weight	Minimum grade	In group	Mandatory	
P1	*Programming assignment (1)*	40%	NO	YES	YES	
P2	*Programming assignment (2)*	40%	NO	YES	YES	
PR	*Exercises*	20%	NO	NO	NO	
OR1	*Oral Presentation and questions (1)*		NO	YES*	YES	
OR2	*Oral Presentation and questions (2)*		NO	YES*	YES	

FinalGrade = 0,4*P1 + 0,4*P2 + 0,2*PR
The oral presentations will be used to validate the work performed on the programming assignments, and an insufficient presentation can downgrade the grade of the programming assignment up to a 50%
* Each member of the group will perform a different part of the presentation, and individual questions will have to be answered

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*