UNIVERSIDAD DE CASTILLA - LA MANCHA

GUÍA DOCENTE

1. General information

Course: PROGRAMMING FUNDAMENTALS I
Code: 42302
Type: BASIC
ECTS credits: 6
Degree: 346 - DEGREE IN COMPUTER SCIENCE AND ENGINEERING
Academic year: 2019-20
Center: 604 - SCHOOL OF COMPUTER SCIENCE AND ENGINEERING (AB)
Group(s): 10 11 12 14 13
Year: 1
Duration: First semester
Main language: Spanish
Second language: English
Use of additional languages: Spanish will be used in traditional group and English will be used in the bilingual group
English Friendly: N
Web site:
Bilingual: Y

Lecturer: **MIGUEL ANGEL GALDON ROMERO** - Group(s): **10 11 12**

Building/Office	Department	Phone number	Email	Office hours
ESII / 1.A.2	SISTEMAS INFORMÁTICOS	2442	miguel.galdon@uclm.es	Consultar Campus Virtual

Lecturer: **ARTURO SIMON GARCIA JIMENEZ** - Group(s): **13**

Building/Office	Department	Phone number	Email	Office hours
ESII / 0.A.4	SISTEMAS INFORMÁTICOS	97144	arturosimon.garcia@uclm.es	Consultar https://www.esiiab.uclm.es/tutorias.php

Lecturer: **ANA AMELIA GONZALEZ LOPEZ** - Group(s): **10 11 12**

Building/Office	Department	Phone number	Email	Office hours
ESII / 1.C.4	SISTEMAS INFORMÁTICOS	2459	ana.gonzalez@uclm.es	Consultar https://www.esiiab.uclm.es/tutorias.php

Lecturer: **JUAN JOSE PARDO MATEO** - Group(s): **10 11 12**

Building/Office	Department	Phone number	Email	Office hours
ESII / 0.A.10	SISTEMAS INFORMÁTICOS	2044	juanjose.pardo@uclm.es	Consultar https://www.esiiab.uclm.es/tutorias.php

2. Pre-Requisites

This subject is taught in the first four-month period of the first year of the Computer Science degree. Because of this, it will be new to most students and it will most likely be their first contact with programming. Therefore, it does not seem logical to stablish prerequisites in this case.

In spite of this, and with the main purpose of guaranteeing the assimilation of the contents and the acquisition of skills of this subject, the student is advised to take advantage of certain personal skills and abilities that they acquired during their primary and secondary studies. Among them, we highlight the critical reading of the texts of the bibliography, the use of the electronic material of this subject available on the virtual campus platform and the active search for complementary material on the Internet.

3. Justification in the curriculum, relation to other subjects and to the profession

Programming Fundamentals I is not an isolated subject, but a part of the curriculum with a close relationship to the rest of the subjects. Thus, this subject has been included in the groups of subjects dedicated to Programming together with Programming Fundamentals II, Data Structures, Programming Methodology and Concurrent and Real Time Programming. Since Programming Fundamentals I is the first subject of the group, it will be one of the fundamental pillars in which the basic concepts of programming will be established, which will subsequently be used by the other subjects of the group.

In addition, the knowledge and skills acquired with this subject will be important for the proper development of other subjects, such as Software Engineering.

Going beyond the university environment and thinking about the future employment of our students, this subject (and all those of that make up the group) will provide them with the necessary skills and abilities to make a good project planning and an evaluation the different alternatives proposed. All this considering that a graduate in computer science is not called to be a mere programmer but to be responsible for large projects.

4. Degree competences achieved in this course

Course competences

Code	Description
BA4	Basic knowledge about the uses and programming of computers, operating systems, data bases, and digital programmes with applications in engineering.
BA5	Knowledge about the structure, organization, functioning, and inter connexions of digital programmes, with their application in engineering problems.
CO7	Knowledge, design, and efficient use of types of data and structures which arise as most appropriate in problem solving.
CO8	Ability to analyse, design, build and maintain applications in a strong, safe, and efficient manner by selecting the most appropriate paradigms and programming languages.
INS1	Analysis, synthesis, and assessment skills.

5. Objectives or Learning Outcomes

Course learning outcomes

Description

Application of basic principles of structured design, led to objects for problem solving.
Resolution of problems throughout basic techniques of algorithm design.

Additional outcomes

Acquire information autonomously, explain it to classmates making sure they have assimilated it.
Code, tune and execute simple programs written in the C programming language.
Know how to choose and manipulate the right types of data for a correct representation of the information.

6. Units / Contents

Unit 1: Introduction to Programming
Unit 2: Representing simple data in memory
Unit 3: Data input/output
Unit 4: Control statements
- **Unit 4.1** Blocks of execution
- **Unit 4.2** Conditional control statements
- **Unit 4.3** Repetition/looping structures

Unit 5: Subprograms
Unit 6: Vectors and Matrices
Unit 7: User defined datatypes
- **Unit 7.1** Registers
- **Unit 7.2** Definition of new datatypes

Unit 8: Data input/output: Files

7. Activities, Units/Modules and Methodology

Training Activity	Methodology	Related Competences (only degrees before RD 822/2021)	ECTS	Hours	As	Com	R	Description
Class Attendance (theory) [ON-SITE]	Combination of methods	BA4 BA5 CO7 CO8	0.66	16.5	N	-	-	
Laboratory practice or sessions [ON-SITE]	Practical or hands-on activities	BA4 BA5 CO7 CO8 INS1	0.66	16.5	N	-	-	
Problem solving and/or case studies [ON-SITE]	Problem solving and exercises	BA4 BA5 CO7 CO8 INS1	0.72	18	N	-	-	
Progress test [ON-SITE]	Assessment tests	BA4 BA5 CO7 CO8 INS1	0.24	6	Y	N	Y	
Study and Exam Preparation [OFF-SITE]	Self-study	BA4 BA5 CO7 CO8 INS1	2.32	58	N	-	-	
Final test [ON-SITE]	Assessment tests	BA4 BA5 CO7 CO8 INS1	0.16	4	Y	Y	Y	
Writing of reports or projects [OFF-SITE]	project-based learning	BA4 BA5 CO7 CO8 INS1	0.8	20	Y	N	Y	
Other off-site activity [OFF-SITE]	Self-study	BA4 BA5 CO7 CO8 INS1	0.44	11	N	-	-	
		Total:	**6**	**150**				
		Total credits of in-class work: 2.44						**Total class time hours: 61**
		Total credits of out of class work: 3.56						**Total hours of out of class work: 89**

As: Assessable training activity
Com: Training activity of compulsory overcoming
R: Rescheduling training activity

8. Evaluation criteria and Grading System

	Grading System		
Evaluation System	**Face-to-Face**	**Self-Study Student**	**Description**
Other methods of assessment	15.00%	0.00%	[INF] A programming project, made in group of 3, will be assessed in this part
Progress Tests	45.00%	0.00%	[ESC] Progress test will be assessed and the final test, if students do this.
Assessment of active participation	10.00%	0.00%	[INF] Participation in classes and the delivery of some extra exercise will be graded.
Test	30.00%	0.00%	[LAB] This part values the laboratory progress test and the laboratory final test if the students do this. These tests will be different for students that attend practical classes and for students who don't attend the practical classes.
Total:	**100.00%**	**0.00%**	

Evaluation criteria for the final exam:

Students must have an average grade of 50%, at least, in the block theorical test + practical test, and have an average grade of 5 to pass the course.

So, the following equation must be true:

"Progress test"*0.45 + "Practical test"*0.3 >= 3.75 (up to 7.5)
and
"Progress test "*0.45 + "Practical test"*0.3 + "project"*0.15 + "participation"*0.1 >= 5 (up to 10)

Students who pass the course through progress tests, laboratory practices and project delivery, will not have to take the final test.

Students not passing the course will take a final test that replace the progress test taken during the continuous evaluation. The final test will have two parts, one written (theoretical) and one in laboratory (practical) with the weights indicated above (45% and 30% respectively). Thus, the total value of this test will be 75% of the final grade. Students are free to decide which part of the exam (written, practical or both) they will take, retaining the grade obtained in the parts that they decide not to take the exam.

In this call, the mark of the programming project will be the one obtained in the delivery made during the course.

The grade corresponding to the achievement in class (5% of the grade) cannot be recovered.

The student who does not pass the mandatory activities/assignments required in the subject will have a mark not higher than 4.00 even if the obtained average was another, including more than 5.00.

Specifications for the resit/retake exam:

Students will have to take a final test that will have two parts, one written (theoretical) and one laboratory (practical) with the weights indicated above (45% and 30% respectively). Thus, the total value of this test will be 75% of the final grade. As in the previous call, students will be able to decide which part of the exam they will take, retaining the grade obtained in previous calls for the part they are not taking in this exam.

Students will be able to retake the part of the programming project if they failed or did not presented it. They will have to carry out and present a new project in this call, which will be different from the one proposed during the course.

It is not possible to bring up the grade corresponding to the achievement in class (10% of the grade). Therefore, the one obtained during the course will be used for the final grade.

The student who does not pass the mandatory activities/assignments required in the subject will have a mark not higher than 4.00 even if the obtained average was another, including more than 5.00.

Specifications for the second resit / retake exam:

The students will take a final test (this test could have a written part and another practical part) whose assessment will be 100%.

9. Assignments, course calendar and important dates

Not related to the syllabus/contents

Hours	**hours**
Progress test [PRESENCIAL][Assessment tests]	6
Final test [PRESENCIAL][Assessment tests]	4

General comments about the planning: This course schedule is APPROXIMATE. It could vary throughout the academic course due to teaching needs, bank holidays, etc. A weekly schedule will be properly detailed and updated on the online platform (Virtual Campus). Note that all the lectures, practice sessions, exams and related activities performed in the bilingual groups will be entirely taught and assessed in English. Classes will be scheduled in three weekly sessions of 1.5 hours over 13 weeks. Evaluation or make-up activities could be performed in the afternoon, in case of necessity.

Unit 1 (de 8): Introduction to Programming

Activities	**Hours**
Class Attendance (theory) [PRESENCIAL][Combination of methods]	1.5
Study and Exam Preparation [AUTÓNOMA][Self-study]	2
Other off-site activity [AUTÓNOMA][Self-study]	1

Unit 2 (de 8): Representing simple data in memory

Activities	**Hours**
Class Attendance (theory) [PRESENCIAL][Combination of methods]	1.5
Laboratory practice or sessions [PRESENCIAL][Practical or hands-on activities]	1.5
Problem solving and/or case studies [PRESENCIAL][Problem solving and exercises]	1
Study and Exam Preparation [AUTÓNOMA][Self-study]	2
Other off-site activity [AUTÓNOMA][Self-study]	1

Unit 3 (de 8): Data input/output

Activities	**Hours**
Class Attendance (theory) [PRESENCIAL][Combination of methods]	1.5
Laboratory practice or sessions [PRESENCIAL][Practical or hands-on activities]	1.5
Problem solving and/or case studies [PRESENCIAL][Problem solving and exercises]	1
Study and Exam Preparation [AUTÓNOMA][Self-study]	2
Writing of reports or projects [AUTÓNOMA][project-based learning]	1
Other off-site activity [AUTÓNOMA][Self-study]	1

Unit 4 (de 8): Control statements

Activities	**Hours**
Class Attendance (theory) [PRESENCIAL][Combination of methods]	3
Laboratory practice or sessions [PRESENCIAL][Practical or hands-on activities]	3
Problem solving and/or case studies [PRESENCIAL][Problem solving and exercises]	3.5
Study and Exam Preparation [AUTÓNOMA][Self-study]	18
Writing of reports or projects [AUTÓNOMA][project-based learning]	5
Other off-site activity [AUTÓNOMA][Self-study]	2

Unit 5 (de 8): Subprograms

Activities	**Hours**
Class Attendance (theory) [PRESENCIAL][Combination of methods]	2.5
Laboratory practice or sessions [PRESENCIAL][Practical or hands-on activities]	3
Problem solving and/or case studies [PRESENCIAL][Problem solving and exercises]	3.5
Study and Exam Preparation [AUTÓNOMA][Self-study]	20
Writing of reports or projects [AUTÓNOMA][project-based learning]	4
Other off-site activity [AUTÓNOMA][Self-study]	2

Unit 6 (de 8): Vectors and Matrices	
Activities	**Hours**
Class Attendance (theory) [PRESENCIAL][Combination of methods]	3
Laboratory practice or sessions [PRESENCIAL][Practical or hands-on activities]	4.5
Problem solving and/or case studies [PRESENCIAL][Problem solving and exercises]	3
Study and Exam Preparation [AUTÓNOMA][Self-study]	5
Writing of reports or projects [AUTÓNOMA][project-based learning]	5
Other off-site activity [AUTÓNOMA][Self-study]	3

Unit 7 (de 8): User defined datatypes	
Activities	**Hours**
Class Attendance (theory) [PRESENCIAL][Combination of methods]	2.5
Laboratory practice or sessions [PRESENCIAL][Practical or hands-on activities]	3
Problem solving and/or case studies [PRESENCIAL][Problem solving and exercises]	4
Study and Exam Preparation [AUTÓNOMA][Self-study]	8
Writing of reports or projects [AUTÓNOMA][project-based learning]	5
Other off-site activity [AUTÓNOMA][Self-study]	1

Unit 8 (de 8): Data input/output: Files	
Activities	**Hours**
Class Attendance (theory) [PRESENCIAL][Combination of methods]	1
Problem solving and/or case studies [PRESENCIAL][Problem solving and exercises]	2
Study and Exam Preparation [AUTÓNOMA][Self-study]	1

Global activity	
Activities	**hours**
Problem solving and/or case studies [PRESENCIAL][Problem solving and exercises]	18
Class Attendance (theory) [PRESENCIAL][Combination of methods]	16.5
Laboratory practice or sessions [PRESENCIAL][Practical or hands-on activities]	16.5
Study and Exam Preparation [AUTÓNOMA][Self-study]	58
Progress test [PRESENCIAL][Assessment tests]	6
Final test [PRESENCIAL][Assessment tests]	4
Writing of reports or projects [AUTÓNOMA][project-based learning]	20
Other off-site activity [AUTÓNOMA][Self-study]	11
Total horas:	150

10. Bibliography and Sources

Author(s)	Title/Link	Publishing house	Citv	ISBN	Year	Description
Herbert Schildt	C: The Complete Reference	McGraw-Hill			2000	4th Edition
Gottfried B.	Programación en C	McGraw- Hill			2005	
Martinez Gil, F.A, Martin Quetglás, G	Introducción a la programación Estructura en C	Universitat de Valencia			2003	
Schildt, H.	C: Manual de Referencia.	McGraw- Hill.				
Carretero, J., García, F., Fernández, J., Calderón, A.	El Lenguaje de Programación C. Diseño e Implementación de Programas	Prentice Hall			2001	
Byron S. Gottfried	Programming with C	McGraw-Hill			1996	