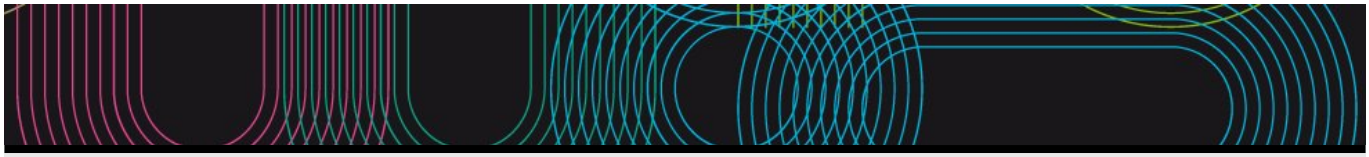

IDENTIFYING DATA				
Programming I				
Subject	Programming I			
Code	V05G301V01105			
Study programme	Degree in Telecommunications Technologies Engineering			
Descriptors	ECTS Credits	Choose	Year	Quadmester
	6	Mandatory	1st	1st
Teaching language	#EnglishFriendly Spanish Galician			
Department				
Coordinator	Rodríguez Hernández, Pedro Salvador			
Lecturers	Blanco Fernández, Yolanda Fernández Masaguer, Francisco Gil Solla, Alberto López Bravo, Cristina Rodríguez Hernández, Pedro Salvador Sousa Vieira, Estrella Suárez González, Andrés			
E-mail	pedro.rodriguez@uvigo.es			
Web	http://faitic.uvigo.es			
General description	The aim of the course is to provide students with basic skills to program in a high level language. The programming paradigm followed is that of "structured programming". English Friendly subject: International students may request from the teachers: a) materials and bibliographic references in English, b) tutoring sessions in English, c) exams and assessments in English.			

Competencies	
Code	
B4	CG4: The ability to solve problems with initiative, to make creative decisions and to communicate and transmit knowledge and skills, understanding the ethical and professional responsibility of the Technical Telecommunication Engineer activity.
B9	CG9: The ability to work in multidisciplinary groups in a Multilanguage environment and to communicate, in writing and orally, knowledge, procedures, results and ideas related with Telecommunications and Electronics.
C6	CE6/T1: The ability to learn independently new knowledge and appropriate techniques for the conception, development and exploitation of telecommunication systems and services
C12	CE12/T7: The knowledge and use of basics in telecommunication networks, systems and service programming.
D2	CT2 Understanding Engineering within a framework of sustainable development.
D4	CT4 Encourage cooperative work, and skills like communication, organization, planning and acceptance of responsibility in a multilingual and multidisciplinary work environment, which promotes education for equality, peace and respect for fundamental rights.

Learning outcomes	
Expected results from this subject	Training and Learning Results
Express the solution of a simple problem by means of algorithms using top-down design.	C12
Identify the data needed to solve a problem and associate them with appropriate datatypes based on their features (size, range, associated operators)	C12
Code simple algorithms using the basic types of statements: assignment, selection and iteration.	C12
Declare and define functions with a proper use of parameters.	C12
Handle I/O operations and file management.	C12
Define and use structured data types.	C12

Define and manage dynamic data structures (lists, stacks, queues and trees).		C12	
Create modules and library functions and use them in programs.		C6 C12	
Predict the result of a sequence of statements, knowing the input data.		C12	
Handle basic tools in an integrated development environment: text editor, compiler, linker, debugger and documentation tools.		C6	
Develop a small scale project following all the phases: requirements analysis, design, implementation, testing and documentation.	B4 B9	C6 C12	D2 D4

Contents

Topic	
Lecture 1: The algorithm and the programming languages.	1. A computer's structure and operation 2. How the program gets into the computer 3. C Programming language 4. The process of developing programs 5. Simple Programming Examples 6. Software engineering concepts
Lecture 2: Grammar and basic elements of C language.	1. Basic elements of a C program 2. Identifiers 3. Expressions 4. Declaration and initialization 5. The assignment statement 6. Formatted input/output
Lecture 3: Iteration and selection statements	1. Control statements 2. Decision statements: (a) if statement (b) if-else statement (c) switch statement 3. Iteration statements: (a) do-while statement (b) while statement (c) for statement 4. Statements for altering the control flow: break and continue statements
Lecture 4: Arrays and pointers	1. Data Structures 2. Arrays: (a) One-dimensional arrays (b) Two-dimensional arrays 3. Strings 4. Pointers: (a) Pointer arithmetic (b) Arrays and pointers (c) Pointers to pointers
Lecture 5: Functions	1. Function declaration and definition 2. Functions with no parameters 3. C inter function communication: local, global and static variables 4. Functions with parameters by value 5. Functions with parameters by reference 6. Command line arguments
Lecture 6: Files	1. Introduction: Types of files 2. Text files in C 3. Declaration 4. File opening and closing 5. File management 6. Operations on characters 7. Operations on strings 8. Formatted operations
Lecture 7: Structured type variables	1. Introduction: Structured data types 2. Structures: (a) Declaration (b) Operations (c) Pointers and structures (d) Structures as parameters
Lecture 8: Lists	1. Introduction: the need for dynamic data structures 2. Dynamic data structures 3. Linked lists (a) Types (b) Most common operations

Planning

	Class hours	Hours outside the classroom	Total hours
Introductory activities	2	0	2
Lecturing	22	22	44
Laboratory practical	14	14	28
Project based learning	8	20	28
Laboratory practice	5	13	18
Objective questions exam	4	20	24
Problem and/or exercise solving	1	5	6

*The information in the planning table is for guidance only and does not take into account the heterogeneity of the students.

Methodologies

	Description
Introductory activities	Introduction to theoretical and practical activities.
Lecturing	Professors present the main theoretical contents related to the subject These sessions can include the development of works and programs by the students. Through this methodology the competencies CE12 and CT2 are developed.
Laboratory practical	During the first part of the term the student codifies, compiles and documents simple programs guided by the instructor. Some of these activities can require the submission of a report in order to be evaluated. Through this methodology the competencies CG4, CE12 and CT2 are developed.
Project based learning	In the last part of the term, the student must complete, under the instructor supervision, a low complexity project which can include individual and in group activities. Through this methodology the competencies CG4, CG9, CE6, CE12, CT2 and CT4 are developed.

Personalized assistance

Methodologies	Description
Lecturing	The professors will provide individual attention to the students along the term, solving their doubts and questions. Questions will be answered during the master sessions or during tutorial sessions. The professors will establish timetables for this purpose at the beginning of the term. This schedule will be published on the subject website.
Laboratory practical	The professors will provide individual attention to the students along the term, solving their doubts and questions about the laboratory practises. Questions will be answered during the lab sessions or during tutorial sessions. The professors will establish timetables for this purpose at the beginning of the term. This schedule will be published on the subject website.
Project based learning	The professors will provide individual attention to the students along the term, solving their doubts and questions about the project. Questions will be answered during the supervising sessions or during tutorial sessions. The professors will establish timetables for this purpose at the beginning of the term. This schedule will be published on the subject website.

Assessment

	Description	Qualification	Training and Learning Results		
Project based learning	The student will develop a project in the last weeks of the term, and will submit the C code implementing it. The project will be assessed individually in the final practice test.	20	B4 B9	C6 C12	D2 D4
Laboratory practice	The student will take 3 mid-term practical tests consisting in the development of small programs in the computer. These tests will assess the student's progress with the laboratory practices.	30	B4	C12	
Objective questions exam	The student will take 3 mid-term theory tests that may consist of: - short answer questions - multiple choice questions These exams will assess individually the student's mastership of the concepts introduced in the master sessions. The final theory exam on the whole contents of the subject will contain this type of questions too.	40	B4	C12	
Problem and/or exercise solving	The final theory exam will have a part consisting of problem and/or exercise solving	10	B4	C12	

Other comments on the Evaluation

The **course planning in lectures** and the estimated time of the **most important assessment milestones** is detailed below (the dates provided for both the theory and the laboratory tests are tentative: the schedule of the midterm/intermediate exams will be approved in the Comisión Académica de Grado (CAG) and will be available at the beginning of each academic semester).

- Week 1: Theory introduction + Lecture 1
- Week 2: Lecture 2 | Practice introduction
- Week 3: Lecture 3 | Practice 1
- Week 4: Lecture 3 | Practice 2
- Week 5: Lecture 4 | Practice 3
- Week 6: Lecture 4 + **Theory Test 1** (PT1) | **Laboratory Test 1** (PL1)
- Week 7: Lecture 5 | Practice 4
- Week 8: Lectures 5 and 6 | Practice 5
- Week 9: Lecture 7 + **Theory Test 2** (PT2) | **Laboratory Test 2** (PL2)
- Week 10: Lecture 8 | Practice 6
- Week 11: Lecture 8 | Practice 7
- Week 12: **Theory Test 3** (PT3) | Project (2h) + **Laboratory Test 3** (PL3)
- Week 13: Project (2h)
- Week 14: Project (2h)
- Week 15: Project (2h)
- Before the final exams, project submission
- Finals: **Final Theory Test** (ETF) - **Final Practice Test** (EPF)

The Final Theory Test (ETF) is an exam that may consist of short answer questions and/or multiple-choice questions and problems and/or exercises. It assesses the student□s command of the contents introduced in the lectures.

The Final Practice Test (EPF) assesses the proper coding in C to deal with a medium level project. While the project development is a group activity, it is assessed individually. Indirectly, the EPF also assesses the student□s command of the contents introduced in the lectures and the laboratory practices.

Following the guidelines of the degree, students are offered two evaluation modes: **continuous evaluation** and **exam-only evaluation**.

The subscription to perform the second mid-term tests, Theory Test 2 (PT2) and / or Laboratory Test 2 (PL2) will be interpreted as the decision to opt for continuous evaluation. The non-enrolment in the second mid-term tests will be interpreted as the decision to opt for the exam-only evaluation.

CONTINUOUS EVALUATION

The continuous evaluation will be considered as "passed" if the final grade (NFC) obtained by the student is at least 5. This final grade is the weighted geometric mean of the mid-term and final tests grades, calculated as follows:

NFC = NPP^0.6 * ETF^0.2 * EPF^0.2

where:

- NPP is the Mid-term Tests Grade, calculated as the weighted arithmetic mean of all the mid-term tests, according to the following expression:

NPP = (NP1 + 2*NP2 + 3*NP3) / 6

Where NPi is the i-th mid-term test grade, calculated as the theory and laboratory grades mean:

NPi = (PTi + PLi) / 2

- ETF is the Final Theory Test grade

- EPF is the Final Practice Test grade

Note that the application of geometric mean implies that it is not possible to pass the subject if any of the notes (NPP, ETF or EPF) is zero.

None of the tests in the continuous evaluation mode is repeatable; that is, the instructor has no obligation to reschedule an evaluated activity missed by a student.

The date and procedures for the revision of the grades will be known before the evaluation tests. The students will have the chance of reviewing the grades preferably within two weeks after the evaluation.

EXAM-ONLY EVALUATION

In order to pass the course by the exam-only evaluation mode, the final grade obtained by the student (NFU) must be at least 5.

This mode will consist of the same final tests as the continuous evaluation one (although with different weights), that is, an exam that may consist of short answer questions and/or multiple choice questions and problems and/or exercises (Final Theory Test, ETF) and a practice test that will evaluate the project (Final Practice Test, EPF). The final grade by exam-only evaluation is the weighted geometric mean of the theory and practice grades, calculated as follows:

NFU = ETF^0.5 * EPF^0.5

Both the continuous evaluation grade (NFC) and the exam-only evaluation grade (NFU) will be computed to all students that take the final tests (theory and practice). The final grade will be the higher one.

A "No Present" grade will be granted if no test is taken by the students after the first mid-term tests (PP1 and PL1).

SECOND CALL EVALUATION

University regulations allow students to take an additional test to pass the course (second call evaluation).

In order to pass the course using this second call evaluation, the final grade obtained by the student (NFS) must be at least 5.

This second call evaluation will consist of an exam that may consist of short answer questions and/or multiple-choice questions and problems and/or exercises (Second Call Theory Test: PTS) and a practice test which will include the evaluation of the project (Second Call Practice Test: PPS). The final grade is the weighted geometric mean between the theory and practice grades, calculated as follows:

NFS = NTS^0.5 * NPS^0.5

Where:

- NTS is the Theory Grade by second call Evaluation: if the student takes the Second Call Theory Test, NTS will be the grade obtained in that test:

NTS = PTS

Otherwise, NTS will be the theory grade obtained in his/her first chance evaluation:

NTS = PPT^0.6 * ETF^0.4

Where PPT is the weighted arithmetic mean of the mid-term theory tests:

PPT = (PT1 + 2 * PT2 + 3 * PT3) / 6

- NPS is the Practice Grade by second call Evaluation: if the student takes the Second Call Practice Test, NPS will be the grade obtained in that test:

NPS = PPS

Otherwise, NPS will be the practice grade obtained in his/her first chance evaluation:

NPS = PPL^0.6 * EPF^0.4

Where PPT is the weighted arithmetic mean of the mid-term laboratory tests:

PPL = (PL1 + 2 * PL2 + 3 * PL3) / 6

END OF DEGREE

University regulations allow students who have 3 or less subjects left to graduate to take an extra call for these subjects.

In order to pass the course using the end-of-degree evaluation system, the final grade obtained by the student (NFG) must be at least 5.

This end-of-degree evaluation will consist of an exam that may consist of short answer questions and/or multiple-choice questions and problems and/or exercises (End-of-degree Theory Test: ETG) and a practice test which will include the evaluation of the project (End-of-degree Practice Test: EPG). The final grade is the weighted geometric mean of the theory and practice grades, calculated as follows:

NFG = ETG^0.5 * EPG^0.5

All the mid-term and final grades will only be valid for the term the student is enrolled to, that is, in case the student repeats the subject, he or she will not retain any of the grades of the previous year.

Plagiarism is regarded as serious dishonest behaviour. If any form of plagiarism is detected in any of the tests or exams, the final grade will be FAIL (0), and the incident will be reported to the corresponding academic authorities for prosecution

---

Sources of information

Basic Bibliography

Brian W. Kernighan, Dennis M. Ritchie, **The C Programming Language**, 1995, Prentice Hall, 1983

Brian W. Kernighan, Dennis M. Ritchie, **El Lenguaje de Programación C**, 1995, Prentice Hall, 1983

Manuel Caeiro Rodríguez, Enrique Costa Montenegro, Ubaldo García Palomares, Cristina López Bravo, J, **Practicar Programación en C**, 2014,

Complementary Bibliography

Ignacio Alvarado Aldea, Jose María Maestre Torreblanca, Carlos Vivas Venegas, Ascensión Zafra Cabeza, **100 Problemas Resueltos de Programación en Lenguaje C para Ingeniería**, 2017, Paraninfo, 2017

Learn C Programming, https://www.tutorialspoint.com/cprogramming/, 2019,

Learn C Programming, https://www.programiz.com/c-programming, 2019,

Stephen G. Kochan, **Programming in C**, 2014, 2005

Osvaldo Cairo Battistuti, **Fundamentos de Programación**, 2006,

José Rafael García-Bermejo Giner, **Programación Estructurada en C**, 2008,

James L. Antonakos, Kenneth C. Mansfield Jr., **Programación Estructurada en C**, 2004, 1997

Jorge A. Villalobos S., Rubby Casallas G., **Fundamentos de Programación: Aprendizaje Activo Basado en Casos**, 2006,

Recommendations

Subjects that continue the syllabus

Informatics: Computer Architecture/V05G301V01109
Programming II/V05G301V01110

Other comments

Programming II course continues this course in the second semester of the first year.

Contingency plan

Description

In case of online tuition, the methodologies used and the tests performed will be the same as in the case of in-person tuition.

The only expected modification is that they will be carried out via Remote Camnpus and Faitic, instead of the School classrooms and laboratories.