IDENTIFYING DATA

Programming I

Subject	Programming I
Code	V05G300V01205
Study programme	(*)Grao en Enxeñaría de Tecnoloxías de Telecomunicación

Descriptors	ECTS Credits	Choose	Year	Quadmester
	6	Mandatory	1st	2nd

Teaching language	Spanish
Department	
Coordinator	Pazos Arias, José Juan
Lecturers	García Palomares, Ubaldo Manuel Pazos Arias, José Juan Ramos Cabrer, Manuel Santos Suárez, José Manuel
E-mail	jose@det.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.

Competencies

Code	
A4	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.
A9	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.
A15	CE6/T1: The ability to learn independently new knowledge and appropriate techniques for the conception, development and exploitation of telecommunication systems and services
A21	CE12/T7: The knowledge and use of basics in telecommunication networks, systems and service programing.

Learning aims

Expected results from this subject	Training and Learning Results
Ability to express the solution of a simple problem with algorithms using top-down design.	A4 A21
Ability to identify the data needed to solve a problem and associate them with appropriate datatypes of the language.	A4 A21
Ability to encode simple algorithms with the basic types of instructions: assignment, selection and iteration.	A21
Ability to declare and define functions with proper use of parameters.	A21
Ability to manage the operations of I / O and file operations.	A21
Ability to define and use structured data types.	A21
Ability to define and manage dynamic data structures (lists, stacks, queues and trees).	A21
Ability to use library modules and create new functions.	A15 A21
Ability to analize a sequence of statements	A21
Ability to handle basic tools in an integrated development environment: text editor, compiler, linker, debugger and tools for documentation.	A15
Ability to use basic concepts of software engineering in the formulation of a small scale project.	A4 A9 A15 A21

Contents

Topic	
Topic 1:The computer and programming languages	1. The computer 2. Programming concepts, software and programming paradigms 3. Stages of software development 4. High-level and low-level programming languages 5. Source code and object code 6. Compilers and interpreters 7. The concept of algorithm 8. Flowcharts, seudo code and natural language for the representation of algorithms 9. General structure of a C program
Topic 2: Basic elements (integers, characters, floats and pointers)	1. Tokens in C: data type, identifier, variable, constant, operator and expression 2. Basic data types and attributes: name, type, memory address, size, value, scope, life. 3. Identifiers 4. Arithmetic operations 5. Logical operations 6. Relacional operations 7. Type conversions 8. Declaration and assignment operations 9. Definition and declaration of pointer variables 10. Basic operations on pointers
Topic 3: Control Instructions (assignment, conditional, iterative and input / output)	1. Basic types of control instructions (sequence, selection and repetition) 2. Selection Instructions (if-else, switch, operator ?...:) 3. Iterative instructions (for, while, do-while), break; and continue; instructions 4.Instructions for standard input/output: printf, scanf 5. Compilation Directives
Topic 4: Functions	1. Modular or procedural programming, problem reduction 2. The structured programming paradigm 3. Declaration and definition of functions 4. Functions without parameters 5. Global variables, local and static 6. Parameter communication by value 7. Parameter communication by reference 8. Entering parameters via the Command window: argc, argv [] 9. Recursive functions
Topic 5: Structured data types	1. Data structures (array, struct, union) 2. One dimensional and two dimensional arrays 3. Strings 4. Declaration and use of data structures 5. Typedef declaration 6. Nested Structures 7. Library functions for string management
Topic 6: Files	1. Concept of file and stream 2. Standard Flows 3. Text files and binary files 4. Basic operations on files, opening and closing, reading, writing 5. Access Modes 6. Macros NULL and EOF 7. Input / output libraries 8. Formatted input/output
Topic 7: Dynamic Memory Management	1. Introduction to dynamic memory management 2. Library functions for dynamic memory management 3. Common linked lists: single, double, circular, circular double binary tree 4. Insertion, replacement and deletion of nodes in linked lists 5. Node structure in linked lists 6. Interaction between lists and files

Planning

	Class hours	Hours outside the classroom	Total hours
Introductory activities	2	2	4
Master Session	25	25	50
Laboratory practises	11	11	22

Projects	11	33	44
Group tutoring	0	3	3
Multiple choice tests	0	4	4
Practical tests, real task execution and / or simulated.	3	6	9
Troubleshooting and / or exercises	4	10	14

*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.
Master Session	Plenary sessions that include the realisation of works and programs.
Laboratory practises	During the first weeks of the term the student codifies, compiles and documents programs guided by the instructor. Some of these activities will be evaluated.
Projects	During the second half of the term, the student must complete a medium complexity project, under the instructor supervision, which includes individual and in group activities.
Group tutoring	Both theoretical and practical issues are discussed in small groups with the instructor.

Personalized attention

Methodologies	Description
Master Session	The webpage of the course informs on the prescheduled office hours that students can consult the instructors. This consulting will be devoted to discard doubts arisen in classroom, laboratory activities and the development of the project.
Laboratory practises	The webpage of the course informs on the prescheduled office hours that students can consult the instructors. This consulting will be devoted to discard doubts arisen in classroom, laboratory activities and the development of the project.
Projects	The webpage of the course informs on the prescheduled office hours that students can consult the instructors. This consulting will be devoted to discard doubts arisen in classroom, laboratory activities and the development of the project.

Assessment

	Description	Qualification
Laboratory practises	Assignments and exercises carried out in the laboratory. These exercises evaluate skills CG4 and CE12/T7.	10
Projects	The evaluation will consist of three activities: 1) report describing the design of the project, 2) a laboratory test focusing on the main ideas behind the project design and implementation, 3) a final report pointing out the main features and the flawless execution of the project. This projecto evaluate skills CG4, CG9, CE6/T1 and CE12/T7.	30
Multiple choice tests	(*)Realizarase un cuestionario tipo test ao longo do curso. Nesta proba avaliaranse as competencias CE6/T1 e CE12/T7.	5
Practical tests, real task execution and / or simulated.	The student will implement one programming exercice. These test evaluate skills CE6/T1 and CE12/T7.	10
Troubleshooting and / or exercises	Exercises proposed during the development of the lectures. Final exam. These proofs will evaluate skills G4 and CE12/T7.	45

Other comments on the Evaluation

Below is the planning of the subject by showing lectures and the estimated time of the most important milestones of assessment:

	Theory	Laboratory	Theory Evaluation	Laboratory Evaluation
Week 1	Lecture 1			
Week 2	Lecture 2	Practice 1		
Week 3	Lecture 2 / Lecture 3	Practice 2		Practical assignment 1 (EP1)
Week 4	Lecture 3	Practice 2		
Week 5	Lecture 3	Practice 3		Practical assignment 2 (EP2)
Week 6	Lecture 4	Practice 3	Web Test 1 (CW1)	

Week 7	Lecture 4	PL1		Laboratory Test (PL1)
Week 8	Lecture 4	Project	Classroom Test 1 (PA1)	
Week 9	Lecture 5	Project		Project Design Report (PR)
Week 10	Lecture 6	Project	Web Test 2 (CW2)	
Week 11	Lecture 6	Project		
Week 12	Lecture 7	Project	Classroom Test 2 (PA2)	
Week 13	Lecture 7	PL2		Project Test (PL2)
Week 14	Lecture 7	Project		
Period of examinations			Final proof on all the contents of the subject (PFT)	Delivery and defence of the project developed in the laboratory (PR)

In all courses the School offers two evaluation modes: Continuous evaluation and comprehensive evaluation. The student must opt to the latter one explicitily, no latter than the grade of the first practical evaluation in the course is officially reported (PL1).

The continuous evaluation will be considered as "approved" if the final grade obtained by the student is at least 5. This final grade is the harmonic mean between the theory and practice, calculated as follows:

$N_F = (2*NP* N_T)/ (NP+ N_T)$

NP (Max. 100%) = EP (Max. 20 %)+ PL1 (Max. 20 %)+PL2 (Max. 20%)+PR (Max. 40 %)

N_T (Max. 100%) =CW (Max. 10 %)+PA1 (Max. 10 %)+PA2 (Max. 20 %)+PFT (Max. 60%)

The use of the harmonic means implies that both NPand N_T must be above 3.3 to approve the course. In any case, the harmonic means must no be inferior to 5.

No evaluation of any activity in the continous evaluation mode is repeatable; that is, an instructor is not obliged to reschedule an evaluated activity missed by a student. No evaluation will be enforced after the end of the term the student is enrolled to.

The comprehensive evaluation consists of practical programming exercices, in paper and with computer, and the presentation on a project report.

NP (no present) will be granted:

1. In continuous evaluation when no asssignment after the first is graded (PL1)
2. In comprehensive evaluation when no assignment is graded

University regulations allow students to take an aditional test to approve the course:

- In the continuos evaluation mode the student should approve a theoretical exam, should do a code implementacion on paper and/or computer and design and implement an extension of the project. The student can:
  - Waive the theoretical exam if his/her theory grade is not under 5 (N_T no lower than 50%);
  - Waive the practical test, i.e., project plus programming if the Laboratory grade obtained in the term was at least 50 %.
  - Nonetheless, the student can take, if he/she wishes so, the theoretical and lab exams.
- In the comprehensive evaluation mode the student should approve a theoretical exam, should do a code implementation on paper and/or computer and design and implement a project. Students under continuos evaluation mode can opt for this alternative evaluation mode.

The student that can opt of voluntary form by one or another option will be able to do it until the moment of delivery or realisation of the corresponding proofs.

The note obtained in the continuous evaluation does not save of a course for the following.

In case of detection of plagiarism in any of the works/test realised the qualification will be of suspense (0) and the professors will communicate to the direction of the school the subject so that it take the measures that consider timely. In the case that the plagiarism detect in any of the works/test of continuous evaluation will not allow that the student follow this procedure of evaluation.

Sources of information
Osvaldo Cairo Battistuti, **Fundamentos de Programación**, 2006,
José Rafael García-Bermejo Giner, **Programación Estructurada en C**, 2008,
Brian W. Kernighan & Dennis M. Ritchie, **El Lenguaje de Programación C**, 1995,
James L. Antonakos & Kenneth C. Mansfield Jr., **Programación Estructurada en C**, 2004,
Jorge A. Villalobos S. & Rubby Casallas G., **Fundamentos de Programación: Aprendizaje Activo Basado en Casos**, 2006,
Manuel Caeiro Rodríguez, Enrique Costa Montenegro, Ubaldo García Palomares, Cristina López Bravo, J, **Practicar Programación en C**, 2014,

Web resources

- http://www.Cprogramming.com
- José R. García-Bermejo Giner: http://maxus.fis.usal.es/FICHAS_C.WEB/11xx_PAGS/11xx.html

Recommendations
Subjects that continue the syllabus
Programming II/V05G300V01302

Subjects that it is recommended to have taken before
Informatics: Computer Architecture/V05G300V01103

Other comments
The subject Programming II is a continuation of this subject in the second course.