Teaching Guide				
Identifying Data				2023/24
Subject (*)	Programming I		Code	614G01001
Study programme	Grao en Enxeñaría Informática			
Descriptors				
Cycle	Period	Year	Type	Credits
Graduate	1st four-month period	First	Basic training	6
Language	SpanishEnglish			
Teaching method	Hybrid			
Prerequisites				
Department	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinador	Cedrón Santaeufemia, Francisco Abel	E-mail	francisco.cedron@udc.es	
Lecturers	Boveda alvarez, Maria del Carmen	E-mail	carmen.boveda@udc.es	
	Calviño Padín, Pablo Alejandro		pablo.calvino.padin@udc.es	
	Castiñeiras Galdo, Brais		brais.cgaldo@udc.es	
	Cedrón Santaeufemia, Francisco Abel		francisco.cedron@udc.es	
	Mato Abad, Virginia		virginia.mato@udc.es	
	Munteanu , Cristian Robert		c.munteanu@udc.es	
	Rabuñal Dopico, Juan Ramon		juan.rabunal@udc.es	
Web	campusvirtual.udc.gal			
General description	This subject is an introduction to programming, in which we learn how to solve problems in a structured language. It helps the student to understand basic data types and structures, while laying the groundwork for the correct design of an algorithm. And to build up the fundamental knowledge of programming in a faster and optimal way, it is necessary to use a language that allows the implementation of the acquired knowledge and serves as a basis for the good development of a computer programmer; the C programming language will be used, both for the practices and for the theoretical examples.			

Study programme competences	
Code	Study programme competences
A4	Coñecementos básicos sobre o uso e a programación dos ordenadores, sistemas operativos, bases de datos e programas informáticos con aplicación na enxeñaría.
A5	Coñecemento da estrutura, organización, funcionamento e interconexión dos sistemas informáticos, os fundamentos da súa programación e a súa aplicación para a resolución de problemas propios da enxeñaría.
B1	Capacidade de resolución de problemas
B3	Capacidade de análise e síntese
B4	Capacidade para organizar e planificar
C3	Utilizar as ferramentas básicas das tecnoloxías da información e as comunicacións (TIC) necesarias para o exercicio da súa profesión e para a aprendizaxe ao longo da súa vida.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrontarse.
C7	Asumir como profesional e cidadán a importancia da aprendizaxe ao longo da vida.

Learning outcomes			
Learning outcomes	Study programme competences		
Knowing and understanding the importance of the programming objectives. Knowing the general aspects of programming languages and paradigms. Knowing the pseudocode and syntax of C language in order to be able to describe algorithms and programs. Knowing the steps to follow for building an application and its main components. Knowing the basic data types using C language. Knowing the control structures for structured programming and the differences between them. Knowing all aspects related to the implementation of functions and procedures.	A4 A5	B1 B3 B4	

Knowing and understanding the importance of the programming objectives. Knowing the general aspects of programming languages and paradigms. Knowing the pseudocode and syntax of C language in order to be able to describe algorithms and programs. Knowing the steps to follow for building an application and its main components. Knowing the basic data types using C language. Knowing the control structures for structured programming and the differences between them. Knowing all aspects related to the implementation of functions and procedures.	A4 A5	B1 B3 B4	
Being able to track an algorithm (in pseudocode) or program (C language), explaining what it is generating and finding possible errors. Being able to solve small algorithms and programs. Solving small algorithms and programs starting from low- to moderate-difficulty problems: given the objectives of the program, to choose and use the best data types and structures, the control structures, to decompose and implement the functions and procedures. Using an appropriate programming style. Learning to make good use of identifiers, appropriate comments, the establishment of preconditions and postconditions, and the good design of procedure and function interfaces.	A4 A5	B1 B3 B4	C3 C6 C7
Being able to track an algorithm (in pseudocode) or program (C language), explaining what it is generating and finding possible errors. Being able to solve small algorithms and programs. Solving small algorithms and programs starting from low- to moderate-difficulty problems: given the objectives of the program, to choose and use the best data types and structures, the control structures, to decompose and implement the functions and procedures. Using an appropriate programming style. Learning to make good use of identifiers, appropriate comments, the establishment of preconditions and postconditions, and the good design of procedure and function interfaces.		B1 B3 B4	C3 C6 C7
Independent learning, planning activities to develop, capacity for abstraction, decision making, initiative and participation.		B3 B4	C3 C6 C7
Independent learning, planning activities to develop, capacity for abstraction, decision making, initiative and participation.		B3 B4	C3 C6 C7

Contents	
Topic	**Sub-topic**

1 BASIC CONCEPTS	1.1 Algorithms 1.1.1 Representation of algorithms 1.2 Programs (applications) 1.2.1 Types of programs 1.3 Programming languages 1.3.1 A historical overview 1.3.2 Classification of languages 1.3.3 Most important language instructions 1.3.4 Properties of languages 1.4 Code compilers 1.5 The structure of a program 1.6 Elements of a program 1.6.1 Predefined symbols 1.6.2 Special symbols 1.6.3 Identifiers 1.6.4 Labels 1.6.5 Comments 1.6.6 Directives 1.6.7 Constants 1.6.8 Numbers 1.6.9 Strings 1.6.10 Variables: declaration and initiation 1.6.11 Variables: memory address 1.7 Output and input 1.7.1 Output sentences 1.7.2 Input sentences 1.8 Data types and operators 1.8.1 Data types 1.8.2 Operators 1.8.3 Expressions

2 Control statements	2.1 Sequential flow 2.2 Alternative syntax 2.2.1 Single statement 2.2.2 Multiple statement 2.3 Repetitive statement 2.3.1 Introduction 2.3.2 Variables associated with loops 2.3.3 Types of loops 2.3.4 FOR loop 2.3.5 Equivalence between loops 2.3.6 Errors with loops 2.3.7 Loop design
3 Program structure	3.1 Functions and Procedures 3.1.1 Types of functions and procedures 3.1.2 Value and reference parameters 3.1.3 Protected parameters 3.1.4 Memory management for procedures 3.1.5 Global and local variables 3.1.6 Side Effects 3.2 Recursion 3.2.1 Why recursion 3.2.2 Infinite recursion
4 Simple data structures	4.1 Arrays and Matrix 4.1.1 ARRAY data type 4.1.2 Declaring an Array 4.1.3 Arrays of more than one dimension 4.1.4 Operations with Arrays and Matrix 4.2 Records 4.2.1 Record data type 4.2.2 Record operations 4.3 Strings 4.3.1 Fixed-length strings 4.3.2 Variable-length strings 4.4 Basic Operations on Arrays 4.4.1 Search operations 4.4.2 Sort operations
5 Input / Output	5.1 Files 5.2 Types 5.3 Operations and access modes 5.4 Specific predefined functions and procedures

Planning				
Methodologies / tests	Competencies	Ordinary class hours	Student?s personal work hours	Total hours

Guest lecture / keynote speech	A4 A5 B1 B3 C6 C7	30	30	60
Laboratory practice	A4 A5 B1 B3 B4 C3 C6 C7	20	50	70
Seminar	B4 C3 C6	8	10	18
Personalized attention		2	0	2
(*)The information in the planning table is for guidance only and does not take into account the heterogeneity of the students.				

Methodologies	
Methodologies	**Description**
Guest lecture / keynote speech	In the theory sessions, the teacher describes the objectives and contents of the subject, to give a particular view of the subject to be dealt with and to relate it to others within the subject. Then the corresponding topic is developed in the form of a lecture session, using the technical tools available, emphasizing certain issues in which the student must deepen his self-learning. The aim is for the student to learn how to algorithmize, use basic data structures and solve simple programming problems. It will be used C language. The master sessions can be face-to-face or through computer platforms such as TEAMS. It is also possible to include explanatory videos of different parts of the theoretical contents.
Laboratory practice	In the practice sessions the student will make programs on paper to later codify them in C language, compile them, execute them and check their level of correction. The program statements will be provided sufficiently in advance so that students can make better use of their time. It is the teacher's mission to supervise the code generated by the student in order to solve doubts, correct bad programming styles and correct errors, counting on the fact that the teacher is not an error-seeking compiler.
Seminar	In the seminar sessions, exercises and practices will be carried out with the aim of detecting gaps in the students' knowledge of the subject matter taught up to that point, and giving the necessary explanations and/or references to correct them. The seminar sessions and resolution of doubts can be done in person or through computer platforms such as TEAMS.

Personalized attention	
Methodologies	**Description**
Laboratory practice Seminar Guest lecture / keynote speech	Both in the master sessions and in the practice laboratories and seminar sessions, students will receive personalized attention at different levels depending on the type of class, detecting the level of assimilation and understanding of the topics explained and the practices required to be implemented. The seminar sessions are where the student can get to know the gaps he has, and show him the way to fill them. Students with part-time enrolment must, at the beginning of the course, speak to the teacher(s) in charge of their group.

Assessment			
Methodologies	**Competencies**	**Description**	**Qualification**

Laboratory practice	A4 A5 B1 B3 B4 C3 C6 C7	During the last weeks of the course with practice, a test will be carried out in the laboratory using computers, which will have a maximum value of 3 points over the total grade of the course. It will be necessary for the program to be carried out by the student in the laboratory to compile and execute correctly and completely.	30
Guest lecture / keynote speech	A4 A5 B1 B3 C6 C7	The grade for the course will be the sum of the results of the Continuous Assessment (during the 15 weeks of the course period) and the results of the Final Exam. The mark of CONTINUOUS EVALUATION, valued in 4 points, is divided in two parts: 1.- A written test will be given in the middle of the course, which will be worth 1 point. 2.- In the last weeks of the course with practice, a test is done in the laboratory using computers that will be worth a maximum of 3 points. The FINAL EXAM will consist of several questions or exercises that the student will have to develop in C language, and will have a value of 6 points in the January call. The official exam, both in the first (January) and in the second opportunity (June/July) will consist of several questions or exercises to be developed in C language. This Final Exam in the January call has a maximum value of 6 points, which will be added to the obtained in the Continuous Assessment. In the June/July Final Exam it will have a maximum value of 7 points that will be added to the one obtained in the practical part of the Continuous Assessment.	70

Assessment comments
The final grades will be determined by the continuous assessment grades and the one obtained in the final exam. The final exam will consist of several questions and programming exercises in the language used in the practice sessions. The gender equality office has included the following guidelines in this section: As stated in the different applicable regulations for university teaching, the gender perspective must be incorporated in this matter (non-sexist language will be used, a bibliography of authors of both sexes will be used, the intervention in class of students will be encouraged and students...)Work will be done to identify and modify prejudices and sexist #attitudes and the environment will be influenced to modify them and promote values of respect and equality.Situations of discrimination based on gender must be detected and actions and measures to correct them will be proposed.

Sources of information	
Basic	- Kernighan, Brian W. Englewood Cliffs (1988). The C Programming Language. New Jersey. Prentice Hall - K.N. King (2008). C programming. A modern Approach. Second Edition.. - James L. Antonakos , Kenneth C. Mansfield (2004). Programación estructurada en C. Madrid. Prentice-Hall - Luis Joyanes Aguilar, Ignacio Zahonero Martínez (2005). Programación en C metodología, algoritmos y estructura de datos. Madrid. McGraw-Hill - José R. García-Bermejo Giner (2008). Programación estructurada en C. Pearson - Luis Joyanes Aguilar (2011). Fundamentos de programación : algoritmos, estructuras de datos y objetos. Madrid. McGraw-Hill
Complementary	- Gabriela Márquez, Sonia Osorio, Noemí Olvera (2011). Introducción a la Programación Estructurada en C. Pearson - Andrés Marzal, Isabel García (2017). Introducción a la Programación con C. Publicacions de la Universitat Jaume I. Servei de Comunicació i Publicacions - Luis Joyanes Aguilar (2002). Programación en C. libro de problemas. Madrid. McGraw-Hill

Recommendations
Subjects that it is recommended to have taken before

Subjects that are recommended to be taken simultaneously
Computer Science Preliminaries/614G01002
Subjects that continue the syllabus
Programming II/614G01006
Other comments
The student must keep in mind that he must do a very important self-learning task, following the scheme: Reading, listening, understanding, asking, studying and practicing. Read: Read the topic to be discussed before attending the theoretical sessions. IT IS VERY IMPORTANT!Attend: Attend in class, don't just be present.Understand: Understand what you are told in the theory sessions, and if you don't ask.Ask: Ask everything you don't understand, don't be in doubt.Study: Study after the sessions, to retain your understanding.Practice: Make many programs, those that are asked, suggested, and others on your own, both on paper and on the computer.Programming is a subject that cannot be learned by studying in two days. The student must mature the concepts, make on paper and in the machine many programs, learning also from the errors when making them. It is a subject that, by means of the system of continuous evaluation, can be approved without more than following, in an active way, the rhythm of the different theoretical and practical sessions. You should pay attention to the particular indications of study reinforcement that the teacher may give you.