Teaching Guide				
Identifying Data				2019/20
Subject (*)	Programming II		Code	614G01006
Study programme	Grao en Enxeñaría Informática			
Descriptors				
Cycle	Period	Year	Type	Credits
Graduate	2nd four-month period	First	Basic training	6
Language	SpanishEnglish			
Teaching method	Face-to-face			
Prerequisites				
Department	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinador	Alonso Pardo, Miguel angel	E-mail	miguel.alonso@udc.es	
Lecturers	Alonso Pardo, Miguel angel Barreira Rodriguez, Noelia Bolón Canedo, Verónica Cabrero Canosa, Mariano Javier De Moura Ramos, Jose Joaquim Gómez Rodríguez, Carlos Guijarro Berdiñas, Berta M. Hernandez Pereira, Elena Maria Monroy Camafreita, Juan Paz López, Alejandro Pérez Sánchez, Beatriz Sanchez Maroño, Noelia Vilares Ferro, Jesus	E-mail	miguel.alonso@udc.es noelia.barreira@udc.es veronica.bolon@udc.es mariano.cabrero@udc.es joaquim.demoura@udc.es carlos.gomez@udc.es berta.guijarro@udc.es elena.hernandez@udc.es juan.monroy@udc.es alejandro.paz.lopez@udc.es beatriz.perezs@udc.es noelia.sanchez@udc.es jesus.vilares@udc.es	
Web	moodle.udc.es			
General description	The subject belongs to the block of courses of Languages and Programming of the degree's Fundamental Training Module. It has a strong relationship with the subjects of Databases, Algorithms and Software Design. It is also related to mathematical subjects, especially Discrete Mathematics. Regarding professional profile, many areas of computing require the ability to work with data structures that are studied in this subject.			

Study programme competences / results	
Code	Study programme competences / results
A3	Capacidade para comprender e dominar os conceptos básicos de matemática discreta, lóxica, algorítmica e complexidade computacional e a súa aplicación para a resolución de problemas propios da enxeñaría.
A4	Coñecementos básicos sobre o uso e a programación dos ordenadores, sistemas operativos, bases de datos e programas informáticos con aplicación na enxeñaría.
B1	Capacidade de resolución de problemas
B3	Capacidade de análise e síntese
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.

Learning outcomes			
Learning outcomes	Study programme competences / results		
Understanding the mechanisms of dynamic memory management.	A4	B1	C6
Understanding the mechanisms of abstraction in the design of data structures.	A4	B1 B3	C3 C6

Building specifications, designing the abstract type from them, using appropriate data structures.	A3 A4	B1 B3	C3 C6
Using appropriate data structures and program algorithms to solve real problems.	A3 A4	B1 B3	C3 C6
Assuming the need for a good specification and a good design as steps prior to coding.	A4	B3	C6
Internalizing good programming practices.	A4	B3	

Contents	
Topic	**Sub-topic**
Dynamic Memory Management	Program memory organization. Definition of pointer variables. Dynamic memory allocation and deallocation. Pointer assignment and comparison operations.
Introduction to Abstract Data Types	Abstraction in programming: Concept, Evolution of abstract data types in computer programming, ADT and Object Oriented Programming. Modularity in programming languages. Abstract Data Type (ADT): Definition and concept, Differences between datatype, data structure and ADT, construction of ADT, Advantages of data abstraction.
Lists	Informal specification of List ADT. Implementation of List ADT. Ordered list ADT: specification and implementation. Multilists and multiordered lists: concept, representations and usage.
Stacks	Informal specification of Stack ADT. Implementation of Stack ADT. Application on computer science.
Queues	Informal specification of Queue ADT. Implementation of Queue ADT. Queue variations. Priority queues. Application on computer science.
Trees	Tree definition and terminology. Binary Tree ADT: Informal specification, Implementation. Binary Tree traversals.
Binary Search Trees	Binary Search Trees. AVL Trees.

Planning				
Methodologies / tests	**Competencies / Results**	**Teaching hours (in-person & virtual)**	**Student?s personal work hours**	**Total hours**
Guest lecture / keynote speech	A3 A4 B1 B3	30	30	60
Problem solving	A3 B1 B3 C6	10	14	24
Laboratory practice	A4 B1 B3 C3 C6	20	26	46
Objective test	A3 A4 B1 B3	3	15	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	The teacher will make a brief description of the topics and basic objectives pursued, in order to provide students with an overview of the subject. In addition they will establish relationships with other concepts previously acquired to build a timeline, and set out the recommended bibliography. They will then develop the theoretical contents using the guest lecture methodology.
Problem solving	In order to reinforce the theoretical concepts, practical cases will be presented, which initially will be resolved by the teacher to guide students. As the theoretical development advance, students will solve problems organized into working groups. This activity, as well as discussion and active participation in class, will be assessed as part of the final mark. When the examples used in the classes of problems or theoretical explanations involve coding or pseudocode, they will be developed showing the successive steps of top-down design. The reason is twofold: a) to get the student used to employ this method and b) to avoid being lost in the details of the particular syntax and language features, instead of paying attention to the understanding and design of the solution. Additional exercises will be assigned as extra-classroom activities. The student must solve them and comment/correct them with the teacher during group and/or individual tutoring . The purpose is to encourage the participation of students and promote, as far as possible, open dialogue and evaluation of solutions. After each topic, several self-assessment tests will be provided using virtual teaching resources, so that the students can verify their learning progress.
Laboratory practice	Practical classes require the students to program data structures in a high-level language. Regular delivery milestones will be proposed to encourage continued study. The practical project assignment will detail the nature of the problem to solve and its specifications, which must be strictly observed. Subsequently, the role of the teacher will be to oversee the practice sessions, solving doubts and correcting misunderstandings, bad programming habits and syntax errors, etc.
Objective test	Summative evaluation of the student through a final exam at the end of the semester, which will be very useful for demonstrating whether the student has acquired the skills of abstraction and design of ADTs and is sufficiently trained to use the precise skills to solve practical cases involving the application of such structures.

Personalized attention	
Methodologies	**Description**
Problem solving Laboratory practice Objective test	Lectures, problem-solving sessions and practical sessions will be developed in response to student progress in understanding and assimilation of the contents. Overall progress will be made compatible with specific attention to those students who have more difficulties in the learning task and with additional support to those that present greater ease and wish to increase their knowledge. Individual tutoring should not be used to extend the contents with new concepts, but to clarify the concepts already discussed in class. The teacher will use them as an interaction that allows him to draw conclusions about the degree of assimilation of the subject by students.

Assessment			
Methodologies	**Competencies / Results**	**Description**	**Qualification**
Problem solving	A3 B1 B3 C6	Various practical tasks to perform in small group tutorials will be proposed. The results obtained and the methods applied to reach the solution will be scored. The mark will only be added to the global marks once the course is passed.	10
Laboratory practice	A4 B1 B3 C3 C6	The practical work are mandatory according to the conditions in each problem assignment. Students must pass all the practical assignments to pass the subject.	20
Objective test	A3 A4 B1 B3	Compulsory fulfillment. Students must pass the exam to pass the subject.	80
Others			

Assessment comments

Practical work - Only students with a mark of FAIL or ABSENT in the first opportunity are allowed to deliver practical works according to the practical definition proposed for second opportunity. - According to article 14, paragraph 4 of existing regulations*, all students who plagiarize the work of others or provide a copy of their practical work will be marked with FAIL, and therefore a failing grade. Part-time enrollment - Students with part-time enrollment must submit the assessment activities under the specific conditions and deadlines. The student will have to communicate their situation to teachers. Absent - A student will have the status of "Absent" if he does not attend the exam in the official evaluation period. Advanced opportunity in December - Student evaluation is based only on a written exam. * Normativa de evaluación, revisión y reclamación de las calificaciones de los estudios de grado y máster universitario, aprobadas por Consello de Goberno de la Universidade da Coruña el 19 de diciembre de 2013. http://www.udc.es/export/sites/udc/normativa/_galeria_down/academica/avaliacionrevrecl.pdf

Sources of information	
Basic	- Narasimha Karumanchi (2017). Data Structures and Algorithms Made Easy, 5th Edition. CareerMonk Publications - Ignacio Zahonero y Luis Joyanes Aguilar (2004). Algoritmos y estructuras de datos: Una perspectiva en C. McGraw-Hill - Kyle Loudon (1999). Mastering Algorithms with C. O'Reilly Media
Complementary	- Reema Thareja (2014). Data Structures Using C - Second Edition. Oxford University Press - Aaron M. Tenenbaum,? Yedidyah Langsam &? Moshe J. Augenstein (1989). Data Structures Using C. Prentice Hall - Richard F. Gilberg & Behrouz A. Forouzan (2005). Data Structures: A Pseudocode Approach with C (2nd Ed). Cengage Learning - Ignacio Zahonero y Luis Joyanes Aguilar (2005). Programación en C. Metodología, Algoritmos y Estructura de Datos, 2º Edición. McGraw-Hill - Luis Joyanes Aguilar, Andrés Castillo Sanz, Lucas Sánchez García e Ignacio Zahonero Martínez (2002). Programación en C. Libro de problemas. McGraw-Hill - Ignacio Zahonero, Lucas García Sánchez, Luis Joyanes Aguilar y Matilde Fernández Azuela (2005). Estructuras de datos en C (Serie Schaum). McGraw-Hill Páxina oficial da contorna de desenvolvemento CLion: https://www.jetbrains.com/clion/

Recommendations
Subjects that it is recommended to have taken before

Programming I/614G01001 Computer Science Preliminaries/614G01002 Discrete Mathematics/614G01004
Subjects that are recommended to be taken simultaneously
Subjects that continue the syllabus
Algorithms/614G01011 Databases/614G01013 Programming Paradigms/614G01014 Software Design/614G01015
Other comments

(*)The teaching guide is the document in which the URV publishes the information about all its courses. It is a public document and cannot be modified. Only in exceptional cases can it be revised by the competent agent or duly revised so that it is in line with current legislation.