UNIVERSIDADE DA CORUÑA

Teaching Guide				
Identifying Data				**2023/24**
Subject (*)	Computer Structure		**Code**	614G01012
Study programme	Grao en Enxeñaría Informática			
Descriptors				
Cycle	**Period**	**Year**	**Type**	**Credits**
Graduate	1st four-month period	Second	Obligatory	6
Language	SpanishEnglish			
Teaching method	Face-to-face			
Prerequisites				
Department	Enxeñaría de Computadores			
Coordinador	Darriba López, Diego		**E-mail**	diego.darriba@udc.es
Lecturers	Beceiro Fernández, Bieito		**E-mail**	bieito.beceiro.fernandez@udc.es
	Darriba López, Diego			diego.darriba@udc.es
	Doallo Biempica, Ramon			ramon.doallo@udc.es
	Fraguela Rodriguez, Basilio Bernardo			basilio.fraguela@udc.es
	Rodríguez Álvarez, Gabriel			gabriel.rodriguez@udc.es
	Sanjurjo Amado, Jose Rodrigo			jose.sanjurjo@udc.es
	Teijeiro Paredes, Diego			diego.teijeiro@udc.es
Web				
General description	Computer architecture, organization and design. Introduction to the main performance metrics. Evaluation and optimization of the performance in the building blocks that compound a computer. Introduction to parallel and storage sytems.			

Study programme competences / results	
Code	**Study programme competences / results**
A15	Capacidade de coñecer, comprender e avaliar a estrutura e a arquitectura dos computadores, así como os compoñentes básicos que os conforman.
B1	Capacidade de resolución de problemas
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 / results**		
Know, understand and ability to evaluate the computer structure and architecture, as well as the components that compound them.	A15	B1	C6 C7

Contents	
Topic	**Sub-topic**
1. Performance evaluation	1. Introduction
	2. Definition of performance metrics
	3. Performance evaluation and comparison
	4. Measurement techniques and benchmarks
2. Instruction level parallelism	1. Introduction
	2. Instruction level dependences and parallelism
	3. Hazards
	4. MIPS pipeline

3. Branch management	1. Static techniques
	2. Dynamic techniques
	3. Branch delay
4. Memory systems	1. Introduction
	2. Main memory
	3. Memory hierarchy
5. Caches	1. Introduction
	2. Operation of the cache system
	3. Cache performance metrics
	4. Optimization techniques
6. Virtual memory	1. Introduction
	2. Pagination
	3. Segmentation
7. Storage systems	1. Basics
	2. Types of storage systems
	3. RAID
8. Buses: connection of I/O and CPU/Memory	1. Introduction
	2. Buses and interconnection
	3. Examples of standard buses

Planning				
Methodologies / tests	Competencies / Results	Teaching hours (in-person & virtual)	Student?s personal work hours	Total hours
Guest lecture / keynote speech	A15	29	37	66
Problem solving	A15 B1	10	20	30
Laboratory practice	A15 C6	20	30	50
Objective test	C7	3	0	3
Personalized attention		1	0	1
(*)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	This type of sessions are master classes complemented with the usage of audiovisual media and the introduction of debating with students phases. The objective is to transfer knowledge and ease the learning process. There will be presentations about the main contents of the subject. Usually, this type of sessions will be an starting point for other activities related to the same topic. In this type of sessions, it will be promoted the adquisition of knowledge associated to compentence A15.
Problem solving	In this type of classes, the teacher will solve several problems which will reinforce the knowledge acquired in the keynote speeches. This type of session will promote the acquisition of compentences A15 and B1 as they improve the capacity of the student to solve computer architecture problems.
Laboratory practice	This type of sessions propose computer driven activities that reinforce the knowledge acquired in other types of sessions. They will allow the familiarization of the student with practial aspects of the subject. The sessions will be completed with a set of self-evaluation tests which let students to find out if they have acquired the skills associated to a particular session. This type of sessions will promote the acquistion of competence A15, as the laboratory activities requires that the student can solve computer architecture problems. As he has to use its knowledge to solve the problems, it also acquires competence C6.

Objective test	This activity evaluates the knowledge and the capacity acquired by the students in this subject.
	It is a written final exam which includes questions to objectively evaluate students.
	This test checks the acquisition of competence A15.
	In general, all the evaluation activities promote the acquisition of competence C7, as it places value on learning.

Personalized attention	
Methodologies	**Description**
Problem solving Laboratory practice	The personalized attention in the laboratory and the problem solving sessions is important to guide the students in their development and learning process. Besides, this attention will serve to validate and evaluate the work of the students in the different stages of their development. It is also recommended that students attend to tutorials when they need it.

Assessment			
Methodologies	**Competencies / Results**	**Description**	**Qualification**
Problem solving	A15 B1	There will be several tests to evaluate the capacity of the students to solve problems autonomously and creatively.	40
Laboratory practice	A15 C6	There will be several tests to evaluate the capacity of the students to solve practical problems using the tools introduced in the lab sessions.	20
Objective test	C7	It will be checked that the student has acquired the knowledge introduced in the master classes, and that he is able to apply them to practical scenarios.	40
Others			

Assessment comments
In order to pass the subject, the student has to reach at least a 50% of the total grade.
The score for the problem-solving part (40%) will be obtained from two tests taken throughout the course (20% each).
At the first opportunity, an objective test will be conducted, accounting for 40%, covering the theoretical and practical contents not assessed in the continuous evaluation tests.At the second opportunity, the objective test will be similar to that of the first opportunity, with the difference being that those students who have not obtained a 50% score in the problem-solving tests will have to do these exercises again.
The part-time students and those that are allowed by the university to not attend to the classes will make the same evaluation tests and exams as the other students. We will make sure that their schedules are compatible with the period of time within they have to attend to classes.
Any dishonest behavior or cheating during the tests or in the laboratory work, once confirmed, will result in a grade of 0 in the grading opportunity in which it happens. In order to do that, the qualification for the first opportunity will be changed if necessary.

Sources of information	
Basic	- Patterson, D. A. y Hennessy, J. L. (2020). Computer Organization and Design MIPS Edition: The Hardware/Software Interface. Morgan Kaufmann
	- Hennessy, J. L. y Patterson, D. A. (2017). Computer architecture. A quantitative approach. Morgan Kaufmann

Complementary	- Harris, S., & Harris, D. (2021). Digital design and computer architecture. Morgan Kaufmann
	- Stallings, W. (2009). Computer Organization and Architecture: Designing for Performance. Prentice Hall
	- Kernighan, R. (1991). El lenguaje de programación C. Prentice Hall
	- Waldron J. (1999). Introduction to RISC Assembly Language Programming. Addison-Wesley
	- Hamacher, C., Vranesic, Z., Zaky, S. y Manjikian, N. (2011). Computer Organization and Embedded systems. McGraw-Hill
	- F. García, J. Carretero, J. D. García y D. Expósito (2009). Problemas Resueltos de Estructura de Computadores. Paraninfo

Recommendations
Subjects that it is recommended to have taken before
Programming I/614G01001
Fundamentals of Computers/614G01007
Subjects that are recommended to be taken simultaneously
Operating Systems/614G01016
Subjects that continue the syllabus
Concurrency and Parallelism/614G01018
Other comments
As collected in the various applicable regulations for university teaching, the incorporation of a gender perspective should be included in this subject (non-sexist language will be used, bibliography from both genders will be suggested, participation in class of students of both genders will be encouraged, etc.)Efforts will be made to identify and modify sexist, racist, or xenophobic prejudices and attitudes, and influence will be exerted on the environment to change them and promote values of respect and equalitySituations of discrimination on the basis of gender, gender identity, origin, etc. should be detected, and actions and measures will be proposed to correct them

(*)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.