UNIVERSIDADE DA CORUÑA

Teaching Guide					
Identifying Data					2021/22
Subject (*)	Concurrency and Parallelism			Code	614G01018
Study programme	Grao en Enxeñaría Informática				
Descriptors					
Cycle	Period		Year	Type	Credits
Graduate	2nd four-month period		Second	Obligatory	6
Language	SpanishGalicianEnglish				
Teaching method	Hybrid				
Prerequisites					
Department	Ciencias da Computación e Tecnoloxías da InformaciónComputaciónEnxeñaría de Computadores				
Coordinador	Paris Fernandez, Javier	E-mail	javier.paris@udc.es		
Lecturers	Darriba López, Diego Enes Álvarez, Jonatan Fraguela Rodriguez, Basilio Bernardo González Domínguez, Jorge Otero Freijeiro, David Paris Fernandez, Javier Pérez Vila, Miguel Anxo Quintela Carreira, Juan Jose Ramos García, Lucia Sanchez Penas, Juan Jose Touriño Dominguez, Juan Veiga Fachal, Jorge	E-mail	diego.darriba@udc.es jonatan.enes@udc.es basilio.fraguela@udc.es jorge.gonzalezd@udc.es david.otero.freijeiro@udc.es javier.paris@udc.es anxo.pvila@udc.es juan.quintela.carreira@udc.es l.ramos@udc.es juan.jose.sanchez.penas@udc.es juan.tourino@udc.es jorge.veiga@udc.es		
Web	campusvirtual.udc.es				
General description					
Contingency plan	1. Modifications to the contents No changes. 2. Methodologies There will be no change to the teaching methodologies. 3. Mechanisms for personalized attention to students email: daily. Students may contact their teacher through email to ask question about the lectures or the laboratory assignments. teams: daily. Students may ask for a meeting on teams to ask questions about the lectures or the laboratory assignaments. 4. Modifications in the evaluation No changes to the evaluation. *Evaluation observations: No changes. 5. Modifications to the bibliography or webgraphy No changes.				

Study programme competences	
Code	**Study programme competences**
A12	Coñecemento e aplicación dos procedementos algorítmicos básicos das tecnoloxías informáticas para deseñar solucións a problemas, analizando a idoneidade e a complexidade dos algoritmos propostos.
A20	Coñecemento e aplicación dos principios fundamentais e técnicas básicas da programación paralela, concorrente, distribuída e de tempo real.
B3	Capacidade de análise e síntese
C4	Desenvolverse para o exercicio dunha cidadanía aberta, culta, crítica, comprometida, democrática e solidaria, capaz de analizar a realidade, diagnosticar problemas, formular e implantar solucións baseadas no coñecemento e orientadas ao ben común.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrontarse.
C8	Valorar a importancia que ten a investigación, a innovación e o desenvolvemento tecnolóxico no avance socioeconómico e cultural da sociedade.

Learning outcomes			
Learning outcomes	**Study programme competences**		
The student should know basic algorithms and how to apply them to solve problems, analyzing the adequacy and complexity of the proposed concurrent and parallel algorithms.	A12	B3	C4
The student should know how to apply the fundamentals of real time, parallel, concurrent and distributed programming.	A20		C6 C8

Contents	
Topic	**Sub-topic**
T1. Concurrent programming fundamentals	1.1 Concepts 1.1.1 Hardware architectures 1.1.2 Operating Systems 1,1.3 Threads and Processes 1.2 Multiprocess programming (fork/join) 1.3 Multithread programming 1.4 Critical section 1.5 Mutual exclusion 1.6 Atomic instructions 1.7 Condition synchronization 1.8 Semaphores 1.8.1 Mutex 1.8.2 Semaphores 1.9 Deadlock. Prevention, avoidance, recovery 1.10 Starvation 1.11 Communication and synchronization 1.12 Scalability
T2. Concurrent Algorithms	2.1 Producers/consumers. 2.2 Readers/writers 2.3 Dining philosophers 2.4 Shared nothing

T3. Parallel programming principles	3.1 Concepts 3.1.1 Levels of paralellism 3.1.2 Data dependencies 3.2 Message passing model 3.2.1 Basic concepts 3.2.2 Point to point communication 3.2.3 Collective operations 3.3 Analysis of parallel algorithms 3.3.1 Performance measure of parallel algorithms 3.4 Methodology for parallel programming 3.4.1 Task decomposition 3.4.2 Task assignment 3.4.3 Optimization techniques 3.5 Schemes for parallel algorithms 3.5.1 Single Process Multiple Data 3.5.2 Master/slave paradigm
T4. Design of parallel algorithms and applications	4.1 Message passing libraries 4.2 Case of study 4.3 Performance evaluation 4.4 Inclusion of optimization techniques

Planning				
Methodologies / tests	Competencies	Ordinary class hours	Student?s personal work hours	Total hours
Guest lecture / keynote speech	A12 A20 C4 C6 C8	30	45	75
Mixed objective/subjective test	A12 A20 B3 C4 C6	3	0	3
Laboratory practice	A12 A20 B3 C8	16	24	40
Problem solving	B3 C6	10	19	29
Practical test:	A12 A20 B3	2	0	2
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	Lecture with audiovisual reinforcement materials, and questions directed at the students to reinforce the transmission of concepts and improve the learning process.
Mixed objective/subjective test	Written exam with questions about the content of the lectures and the practical problems solved in the laboratory practice.
Laboratory practice	Practical activities aimed at enhancing the comprehension of the material by the students, such as programming exercicies.
Problem solving	Solving of concrete problems that appeared during the laboratory practice, possibly exploring multiple solutions.
Practical test:	Tests about the contents of the laboratory practices. Part of the ongoing evaluation.

Personalized attention	
Methodologies	Description

Laboratory practice Problem solving	During the laboratoy practice, seminars and problem solving sessions students will be able to ask questions about the contents. The teacher, after considering these questions, will reinforce specific topics, solve problems that involve the concepts that are unclear, or any other activity that may help to improve the understanding of the content. All tutoring sessions will be held online.

Assessment			
Methodologies	**Competencies**	**Description**	**Qualification**
Practical test:	A12 A20 B3	Ongoing assesment exams on the contents of the lectures and the laboratory practices.	20
Laboratory practice	A12 A20 B3 C8	Practical exercises divided on two blocks: concurrency and parallelism. Each block is worth 50% of the laboratory practice grade. Exercises can be solved in groups of two, but will be graded individually.	30
Mixed objective/subjective test	A12 A20 B3 C4 C6	Exam on the contents explained during the lectures and practiced in the laboratory. There will be two parts: concurrency (topics T1 and T2) and parallelism (topics T3 and T4). Each part is worth 50% of the grade of the mixed test.	50

Assessment comments
The final grade will be the weighted addition of the mixed test, the laboratory practice grades, and the practical test grades. In order to pass it is necessary to get at least 50% of the maximum grade. For the July evaluation only the mixed test will be graded again (70% of the total grade). The grade obtained during the term in the laboratory practice (30% of the final grade) and the practicas tests (20% of the final grade) will be used for both the June and July evaluations. The grade for the laboratory practices will not be reassesed during the second opportunity. The evaluation of the laboratory practices must be done in the group assigned to each student. No special consideration will be given to students with part time enrollment.

Sources of information	
Basic	- Doug Lea (2000). Concurrent programming in Java design, principles and patterns . Reading, Massachusetts: Addison Wesley - Joe Armstrong (2007). Programming Erlang: Software for a Concurrent World. United States: Pragmatic Programmers - Francisco Almeida [et al.] (2008). Introducción a la Programación Paralela. Madrid: Paraninfo Cengage Learning - Peter S. Pacheco (1997). Parallel Programming with MPI. San Francisco, California : Morgan Kauffman
Complementary	- Wilkinson, B. y Allen, M.. (1999). Parallel Programming. Techniques and Applications Using Networked Workstations and Parallel Computers. . Upper Saddle River, New Jersey : Prentice Hall,

Recommendations
Subjects that it is recommended to have taken before
Programming II/614G01006 Algorithms/614G01011 Computer Structure/614G01012 Programming Paradigms/614G01014 Software Design/614G01015
Subjects that are recommended to be taken simultaneously
Operating Systems/614G01016 Networks/614G01017 Software Process/614G01019
Subjects that continue the syllabus

Internet and Distributed Systems/614G01023
Other comments