UNIVERSIDADE DA CORUÑA

Teaching Guide				
Identifying Data				2018/19
Subject (*)	Programming Language Design		Code	614G01065
Study programme	Grao en Enxeñaría Informática			
Descriptors				
Cycle	Period	Year	Type	Credits
Graduate	1st four-month period	Fourth	Obligatory	6
Language	Spanish			
Teaching method	Face-to-face			
Prerequisites				
Department	Computación			
Coordinador	Alonso Pardo, Miguel angel	E-mail	miguel.alonso@udc.es	
Lecturers	Alonso Pardo, Miguel angel Graña Gil, Jorge Vilares Ferro, Jesus	E-mail	miguel.alonso@udc.es jorge.grana@udc.es jesus.vilares@udc.es	
Web	moodle.udc.es			
General description	This course deals with the following aspects of the specification and design of programming languages: * Design Criteria for control structures and datat ypes. * Design of object-oriented programming languages. * Models for the formal definition of the semantics of programming languages * Formal specification of type systems. Subtyping relations * Computability. Analysis of complexity and its relation to the design of programming languages.			

Study programme competences / results	
Code	Study programme competences / results
A39	Capacidade para ter un coñecemento profundo dos principios fundamentais e modelos da computación, e saber aplicalos para interpretar, seleccionar, valorar, modelar, e crear novos conceptos, teorías, usos e desenvolvementos tecnolóxicos relacionados coa informática.
A40	Capacidade para coñecer os fundamentos teóricos das linguaxes de programación e as técnicas de procesamento léxico, sintáctico e semántico asociadas, e saber aplicalas para a creación, o deseño e o procesamento de linguaxes.
B1	Capacidade de resolución de problemas
C2	Dominar a expresión e a comprensión de forma oral e escrita dun idioma estranxeiro.
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		
To introduce lambda-calculus, typed and untyped, as the fundamental core of programming languages.	A39 A40	B1	C2 C6
To understand the formal base of typing and subtyping systems	A39 A40	B1	C2 C6
To understand and master the design principles of object-oriented languages and the implications that design choices have on the development of programs	A39 A40	B1	C6
To manage the design principles of the main control structures of programming languages and their implications for program development	A39 A40	B1	C6
To manage the design principles of the main data structures of programming languages and their implications for program development	A39 A40	B1	C6

Contents	
Topic	**Sub-topic**
Formal definition of type systems	Operational, denotational and axiomatic semantics An introduction to lambda-calculus Typed lambda-calculus Subtyping
Object-Oriented Languages	Fundamental concepts of object-oriented languages Type problems in object-oriented languages
Principles of Programming Language Design	Names, scopes and binding Control flow Data types Subroutines
Computability and Complexity	Computability and Lambda calculus Complexity classes

Planning				
Methodologies / tests	**Competencies / Results**	**Teaching hours (in-person & virtual)**	**Student?s personal work hours**	**Total hours**
Laboratory practice	A39 C2 C6	14	42	56
Workshop	B1 C6	7	14	21
Objective test	A40 B1	2	6	8
Guest lecture / keynote speech	A40 C2	21	42	63
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**
Laboratory practice	Activity that allows students to learn effectively through the realization of practical activities, in this case lab assigments, demonstrations and exercises.
Workshop	They are made to complement all other activities, in some cases solved independently by the student and sometimes under the supervision of the professor.
Objective test	Test in which the knowledge acquired in the theoretical and practical parts of the subject will be assessed.
Guest lecture / keynote speech	Oral presentation complemented with the use of audiovisual media and the formulation of questions to/by the students, with the aim of transmitting knowledge and stimulate critical thinking

Personalized attention	
Methodologies	**Description**
Workshop Laboratory practice	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**

Workshop	B1 C6	Otras actividades evaluables, cuyo contenido se desarrollarán en las horas de TGR.	20
Objective test	A40 B1	Ejercicio escrito	40
Laboratory practice	A39 C2 C6	Trabajos de laboratorio	40

Assessment comments
The theoretical part of the course computes 40% of the grade. TGR activities compute for 20% of the grade. The evaluation of TGR's will be held in the written exam. The remaining 40% is divided between lab assignments and any other evaluation activities performed throughout the course.To pass the course the student must pass each and every one of the sections of the evaluation. For second chance, the results of each section at the first opportunity will be preserved.In the case of part-time students, failure to attend classes and practices which are duly justified TGR will not be penalized.An student can get bonus points for doing the activities in English (for example, deliver the report of a lab assignment in English, present an exercise in English, etc). In no case he/she will be penalized for performing activities in Spanish and/or Galician.

Sources of information	
Basic	- Michael L. Scott (2009). Programming Language Pragmatics. Third edition. Morgan Kaufmann Publish- ers, Burlington, MA - Kim B. Bruce (2002). Foundations of Object-Oriented Languages: Types and Semantics. The MIT Press, Cambridge, MA - Benjamin C. Pierce (2002). Types and Programming Languages. The MIT Press, Cambridge, MA - Fortnow, Lance (2013). P, NP, and the search for the impossible. Princeton University Press
Complementary	- David A. Watt (2004). Programming Language Design Concepts. ohn Wiley and sons, Chichester, West Suusex, England - Franklyn A. Turbak and David K. Gifford (2008). Design Concepts in Programming Languages. MIT Press, Cambridge, MA - Robert W. Sebesta (2010). Concepts of Programming Languages. Pearson

Recommendations
Subjects that it is recommended to have taken before
Programming Paradigms/614G01014 Theoretical Computer Science/614G01039
Subjects that are recommended to be taken simultaneously
Language Processing/614G01067
Subjects that continue the syllabus
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.