



Teaching Guide

Teaching Guide				
Identifying Data				2020/21
Subject (*)	Programming Paradigms		Code	614G01014
Study programme	Grao en Enxeñaría Informática			
Descriptors				
Cycle	Period	Year	Type	Credits
Graduate	1st four-month period	Second	Obligatory	6
Language	Spanish			
Teaching method	Hybrid			
Prerequisites				
Department	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinador	Graña Gil, Jorge		E-mail	jorge.grana@udc.es
Lecturers	Graña Gil, Jorge Molinelli Barba, Jose Maria Paris Fernandez, Javier Vilares Ferro, Jesus		E-mail	jorge.grana@udc.es jose.molinelli@udc.es javier.paris@udc.es jesus.vilares@udc.es
Web	moodle.udc.es			
General description	Resolución de problemas usando diferentes técnicas de programación: estruturada, orientada a obxectos, declarativa, etc.			
Contingency plan	1. Modifications to the contents 2. Methodologies *Teaching methodologies that are maintained *Teaching methodologies that are modified 3. Mechanisms for personalized attention to students 4. Modifications in the evaluation *Evaluation observations: 5. Modifications to the bibliography or webgraphy			

Study programme competences

Code	Study programme competences
A7	Capacidade para deseñar, desenvolver, seleccionar e avaliar aplicacións e sistemas informáticos que aseguren a súa fiabilidade, seguranza e calidade, conforme a principios éticos e á lexislación e normativa vixente.
A13	Coñecemento, deseño e utilización de forma eficiente dos tipos e estruturas de datos máis adecuados á resolución dun problema.
A14	Capacidade para analizar, deseñar, construír e manter aplicacións de forma robusta, segura e eficiente, elixindo o paradigma e as linguaxes de programación máis adecuados.
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.
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
-------------------	-----------------------------



Cóñecer os fundamentos e principios básicos da programación, incluíndo variables, tipos, expresións, estruturas de control, estruturas de datos e recurrencia.	A7 A13	B1	C6 C8
Empregar e aplicar os diferentes paradigmas de programación para a resolución de problemas.	A7 A14	B1	C6 C8

Contents	
Topic	Sub-topic
Programación Declarativa: Programación Funcional	Tipos e valores. Expresións e definicións. "Pattern-matching". Funcións. Funcións recursivas. Terminación. Recursividade terminal. "Currying". Funcións de orde superior. Tipos parametrizados. Tipos recursivos. Polimorfismo. Transparencia referencial. Excepcións.
Programación imperativa	Estado da máquina. Variables. Asignación. Programación estruturada. Estructuras de control: Composición secuencial, alternativa e iterativa. Procedimentos e funcións. Paso de parámetros por referencia e por valor. Efectos colaterais. Programación imperativa vs. declarativa.
Programación Orientada a Obxectos	Obxectos, atributos e métodos. Clases e herdanza. Polimorfismo. Programación Orientada a Obxectos vs. imperativa. Programación Orientada a Obxectos vs. declarativa.
A linguaxe de programación Objective Caml	Programación funcional, imperativa e orientada a obxectos en Ocaml. Os compiladores de Ocaml. Entrada / Saída. Módulos e librerías. Abstracción, encapsulación e compilación separada. Módulos, interfaces e signaturas.

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



Guest lecture / keynote speech	A7 A13 A14 B1 C6 C8	30	20	50
Objective test	A13 A14 B1	4	16	20
Laboratory practice	A7 A13 A14 B1	20	20	40
Practical test:	A7 A13 A14 B1 C6 C8	10	20	30
Personalized attention		10	0	10
(*)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	Exposición na aula dos contidos básicos da materia.
Objective test	Exame escrito.
Laboratory practice	Exercicios de programación para a posta en práctica do visto nas clases maxistras, con atención personalizada por parte do profesor de prácticas en horario de laboratorio.
Practical test:	Probas prácticas presenciais periódicas, aproximadamente unha de cada tres semanas, realizadas no laboratorio.

Personalized attention	
Methodologies	Description
Laboratory practice Practical test:	Durante o horario de prácticas de laboratorio supervisarase o traballo dos estudantes e asesoraráselles na resolución dos exercicios. Asesorarase persoalmente aos estudantes na preparación das probas prácticas para a súa realización nas horas reservadas para as mesmas. O profesor tentará solucionar aquelas dúbidas que poidan xurdir respecto ao temario da materia.

Assessment			
Methodologies	Competencies	Description	Qualification
Laboratory practice	A7 A13 A14 B1	Asistencia, realización e entrega de prácticas de laboratorio.	20
Objective test	A13 A14 B1	Exame escrito.	60
Practical test:	A7 A13 A14 B1 C6 C8	Coa realización das probas prácticas periódicas poderá consolidarse até un 20% da nota final. A porcentaxe non consolidada pasará a computarse na proba obxectiva. A valoración do exame escrito realizarase pola porcentaxe que reste até o 80%.	20

Assessment comments

Sources of information	
Basic	- WIKSTRÖM, A. (). Functional Programming Using Standard ML. Prentice Hall - John Whittington (2013). OCaml from the very beginning. Coherent Press - Andrei De Araújo Formiga (2015). OCaml: Programação funcional na prática. Casa de Código Manual de Objective CamlManual de Objective Caml



Complementary	- WEIS, P. & LEROY, X. (1993). Le Language Caml. InterEditions - COUSINEAU, G. & MAUNY, M. (1998). The functional Approach to Programming. Cambridge University Press. - John Whittington (2014). More OCaml. Algorithms, Methods & Diversions. Coherent Press - Yaron Minsky, Anil Madhavapeddy & Jason Hickey (2013). Real World OCaml. O'Reilly - PAULSON, L. C. (1991). ML for the Working Programmer. Cambridge University Press. - Michel Quercia (2000). Nouveaux exercices d'algorithmique. Éditions Vuibert, Paris - Philippe Narbe (2005). Programmation fonctionnelle, générique et objet: une introduction avec le langage OCaml. Vuibert, Paris - Jacques Rouablé (1997). Programmation en Caml. Eyrolles, Paris - Luc Albert (1997). Cours et exercices d'informatique. Thomson Publishing International, Paris - Joshua B. Smith (2006). Practical OCaml. Apress DOWNEY, A.; MONJE, N.: Think OCaml. How to Think Like a (Functional) Programmer CHAILLOUX, E.; MANOURY, P. & PAGANO, B.: Developing Applications With Objective Caml. DOWNEY, A.; MONJE, N.: Think OCaml. How to Think Like a (Functional) Programmer CHAILLOUX, E.; MANOURY, P. & PAGANO, B.: Developing Applications With Objective Caml.
----------------------	--

Recommendations

Subjects that it is recommended to have taken before

Programming I/614G01001
Discrete Mathematics/614G01004
Programming II/614G01006

Subjects that are recommended to be taken simultaneously

Algorithms/614G01011
Software Design/614G01015

Subjects that continue the syllabus

Concurrency and Parallelism/614G01018
Intelligent Systems/614G01020

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.