



## Teaching Guide

Identifying Data					2018/19
<b>Subject (*)</b>	Programming Paradigms	<b>Code</b>	614G01014		
<b>Study programme</b>	Grao en Enxeñaría Informática				
Descriptors					
<b>Cycle</b>	<b>Period</b>	<b>Year</b>	<b>Type</b>	<b>Credits</b>	
Graduate	1st four-month period	Second	Obligatory	6	
<b>Language</b>	Spanish				
<b>Teaching method</b>	Face-to-face				
<b>Prerequisites</b>					
<b>Department</b>	Computación				
<b>Coordinador</b>	Molinelli Barba, Jose Maria	<b>E-mail</b>	jose.molinelli@udc.es		
<b>Lecturers</b>	Castro Souto, Laura Milagros Graña Gil, Jorge Molinelli Barba, Jose Maria Paris Fernandez, Javier Vilares Ferro, Jesus	<b>E-mail</b>	laura.milagros.castro.souto@udc.es jorge.grana@udc.es jose.molinelli@udc.es javier.paris@udc.es jesus.vilares@udc.es		
<b>Web</b>	moodle.udc.es				
<b>General description</b>	Resolución de problemas usando diferentes técnicas de programación: estruturada, orientada a obxectos, declarativa, etc.				

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

## 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
Directed discussion	A7 A13 A14 C6 C8	8	0	8
Objective test	A13 A14 B1	4	16	20
Supervised projects	A7 A13 A14 B1	2	20	22
Laboratory practice	A7 A13 A14 B1	20	20	40
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.
Directed discussion	Nas horas de titorías en grupos reducidos, discutirase cos estudantes a formulación e resolución de problemas avanzados.
Objective test	Exame escrito.
Supervised projects	O estudante poderá acordar co profesor a preparación de certos traballos para a súa presentación e discusión con outros estudantes durante as horas de titorías en grupos reducidos.
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.

Personalized attention	
Methodologies	Description
Laboratory practice Supervised projects	<p>Durante o horario de prácticas de laboratorio supervisarase o traballo dos estudantes e asesoraráselles na resolución dos exercicios.</p> <p>Asesorarase persoalmente aos estudantes na preparación dos traballos tutelados para a súa presentación nas horas de titorías en grupos reducidos.</p> <p>O profesor tentará solucionar aquelas dúbidas que poidan xurdir respecto ao temario da materia.</p>

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
Supervised projects	A7 A13 A14 B1	Coa realización de traballos tutelados e a súa defensa e discusión durante as horas de Titorías en Grupos Reducidos 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	<ul style="list-style-type: none"> <li>- WIKSTRÖM, A. (). Functional Programming Using Standard ML. Prentice Hall</li> <li>- John Whittington (2013). OCaml from the very beginning. Coherent Press</li> <li>- Andrei De Araújo Formiga (2015). OCaml: Programação funcional na prática. Casa de Código</li> </ul> <p>Manual de Objective CamlManual de Objective Caml</p>



<b>Complementary</b>	<ul style="list-style-type: none"><li>- WEIS, P. &amp; LEROY, X. (1993). Le Languaje Caml. InterEditions</li><li>- COUSINEAU, G. &amp; MAUNY, M. (1998). The functional Approach to Programming. Cambridge University Press.</li><li>- John Whittington (2014). More OCaml. Algorithms, Methods &amp; Diversions. Coherent Press</li><li>- Yaron Minsky, Anil Madhavapeddy &amp; Jason Hickey (2013). Real World OCaml. O'Reilly</li><li>- PAULSON, L. C. (1991). ML for the Working Programmer. Cambridge University Press.</li><li>- Michel Quercia (2000). Nouveaux exercices d'algorithmique. Éditions Vuibert, Paris</li><li>- Philippe Narbe (2005). Programmation fonctionnelle, générique et objet: une introduction avec le langage OCaml. Vuibert, Paris</li><li>- Jacques Rouablé (1997). Programmation en Caml. Eyrolles, Paris</li><li>- Luc Albert (1997). Cours et exercices d'informatique. Thomson Publishing International, Paris</li><li>- Joshua B. Smith (2006). Practical OCaml. Apress</li></ul> <p>DOWNEY, A.; MONJE, N.: Think OCaml. How to Think Like a (Functional) Programmer CHAILLOUX, E.; MANOURY, P. &amp; PAGANO, B.: Developing Applications With Objective Caml. DOWNEY, A.; MONJE, N.: Think OCaml. How to Think Like a (Functional) Programmer CHAILLOUX, E.; MANOURY, P. &amp; PAGANO, B.: Developing Applications With Objective Caml.</p>
----------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 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.