# Universida<sub>de</sub>Vigo

Subject Guide 2013 / 2014

711111111		-		
		1		
IDENTIEYIN	G DATA			7711111111
(*)Program	ación II			
Subject	(*)Programación II			
Code	V05G300V01302			
Study	(*)Grao en			
programme	Enxeñaría de			
1 3	Tecnoloxías de			
	Telecomunicación			
Descriptors	ECTS Credits	Choose	Year	Quadmester
	6	Mandatory	2nd	1st
Teaching	Spanish			
language				
Department				
Coordinator	Fernández Masaguer, Francisco			
Lecturers	Blanco Fernández, Yolanda			
	Fernández Masaguer, Francisco			
	Servia Rodriguez, Sandra			
E-mail	f_masaguer@yahoo.es			
Web	http://www.faitic.es			
General	The general aim of this subject is to provide the stude	nts with the theor	etical foundation	ns and the practical
	competitions that allow them analyze, design, develop and debug computer applications following the paradigm of oriented objects programming (OOP). This is an essentially practical subject oriented to the work of the students in the development of software projects. To make this task easier, the subject includes an introduction to the Engineering of the Software. In this sense, the focus is not put on all the well-known phases of the processes of development software (ranging from capture and description of requirements to the deployment of the systems), but just on main stages related to analysis, design, implementation and debugging. Firstly, the engineering of the software is presented as an indispensable discipline for the development of big computer applications, showing the main challenges to face and the basic concepts behind them. Next, the elements of the OOP will be detailed by resorting to UML elements and diagrams, which will be used by the students in their developments. To reach this general aim the contents that will be handled in the subject are the following ones: OOP paradigm: basic concepts, classes and objects. Encapsulation. Concepts of decoupling and cohesion Inheritance, abstraction, polymorphism and reuse Relations between classes: Generalization, association and dependency Communication between objects: methods, events, messages Persistence. Storage in files and in databases Generation, capture and processing of exceptions Introduction to the Engineering of the Software Concepts of the Engineering of the Software. Historical review or Introduction and concept of Cycle of Life.			
Competend	ies			
Code				
A6 CG6: T A9 CG9: T	ne aptitude to manage mandatory specifications, proce ne ability to work in multidisciplinary groups in a Multila provide procedures, results and ideas related with	edures and laws. Anguage environm	ient and to com	municate, in writing and
A59 (CE50/ (OOP)·	18)The ability to develop, interpret and debug program classes and objects, encapsulation, relations among classes and objects.	ms using basic cor asses and objects	and inheritance	Oriented Programming
A60 (CE51/	19) The ability of basic application of phases of analys	is, design, implem	nentation and de	bugging of OOP

programs.

A61 (CE52/T20) The ability of manipulation of CASE tools (editors, debuggers).

A62 (CE53/T21) The ability of developing programs considering to the basic principles of software engineering quality taking into account the main existing sources of norms, standards and specifications.

B5 The ability to use software tools to search for information or bibliographical resources

Learning aims		
Expected results from this subject	Train	ing and Learning Results
To understand the fundamental concepts of the Object Oriented Programming model (OOP) and	A9	
carry them to practise iusing the most representative object oriented programing languaje (Java).	A59	
To introduce in the use of the UML language, the ISO standard language for software modeling, for	A6	B5
the making of structure, behaviour and interaction diagrams, and fundamental for the	A61	
documentation in the phases of analysis and design of OO programs.	A62	
To develop skills in the process of analysis, design, implementation and debugging of OOP	A60	·
applications taking into account the main standards and quality norms.	A62	
To adquiry maturity in development and debugging programming techniques to allow the	A62	
autonomous learning of new capacities and programming languages.		
To adquiry familiarity with the use of a modern software development tool (Eclipse) to facilitate the	A60	'
design, development and debugging of programs.	A61	

Contents	
Торіс	
1. Introduction to OO paradigm	a. Brief introduction to the subject and organization.
	b. Birth of the paradigm
	c. Bases: classes and objects
	d. Concepts of encapsulation, inheritance (generalization), and
	polymorphism
	e. Brief Introduction to UML
2. Encapsulation	<ul> <li>Classes, interfaces and packages</li> </ul>
	b. Methods and variable member. Visibility. Resolution of field.
	c. Method constructor
	d. Step of parameters: pointers and references
	e. Pointers to objects
3. Inheritance	<ul> <li>Derived classes and types of inheritance</li> </ul>
	b. Abstract Classes
	c. Multiple Inheritance
	d. Object class
<ol> <li>Object-Oriented design</li> </ol>	a. Design Basics
	b. Use of UML diagrams
5. Polymorphism	a. Overloading and overwriting
	b. Abstract classes and interfaces
	c. Generic classes
6. Exception Handling	a. Exception Basics
	b. Handling Java exceptions
7. Recursion	a. Recursive void methods
	b. Recursive methods that return a value
	c. Thinking recursively

Planning			
	Class hours	Hours outside the classroom	Total hours
Master Session	28	42	70
Troubleshooting and / or exercises	9	9	18
Presentations / exhibitions	1	1	2
Autonomous troubleshooting and / or exercises	5	10	15
Projects	7	31	38
Practical tests, real task execution and / or simulated.	2	0	2
Case studies / analysis of situations	0	1	1
Troubleshooting and / or exercises	2	0	2
Practical tests, real task execution and / or simulated.	2	0	2
*The information in the planning table is for guide	nea anly and dage ne	t take into account the het	are gone it , of the students

\*The information in the planning table is for guidance only and does not take into account the heterogeneity of the students.

 Methodologies

 Description

 Master Session
 Classes that combines explanation of theoretical concepts and realization of exercises. These exercises can be either resolved by the professor or by the students individually or in groups. The goal is to promote the debate and reinforce the acquisition of skills.

Troubleshooting and / or	In the computer rooms, the professor will pose challenges to be resolved by the students, thus
exercises	discussing collectively the possible options to face a solution.
Presentations / exhibitions	The students will present in the computer room the design proposed to resolve the software project defined by the professor during the second part of the course.
	The goal is to compare the different proposals, identifying advantages and possible deficiencies that will serve as feedback to face the final design.
Autonomous troubleshooting and / or exercises	Students individually will resolve the problems posed by the professor in the computer room. Solutions and doubts that arise in addressing these problems will be put together to agree the best way to fix each concern.
Projects	The students will develop a software project defined by the professor. The development of this project will require face-to-face work in the computer room (supported by the professor) and individual work.

Personalized attention	
Methodologies	Description
Troubleshooting and / or exercises	The personalized attention will consist of following-up the work of each student, tracking the solutions proposed for each problem posed in the sessions in the computer room and the exhibition of their UML designs for the proposed software project.
Presentations / exhibitions	The personalized attention will consist of following-up the work of each student, tracking the solutions proposed for each problem posed in the sessions in the computer room and the exhibition of their UML designs for the proposed software project.
Projects	The personalized attention will consist of following-up the work of each student, tracking the solutions proposed for each problem posed in the sessions in the computer room and the exhibition of their UML designs for the proposed software project.
Autonomous troubleshooting and / or exercises	The personalized attention will consist of following-up the work of each student, tracking the solutions proposed for each problem posed in the sessions in the computer room and the exhibition of their UML designs for the proposed software project.

Assessment		
	Description	Qualification
Projects	The students, organized into groups of 2 people, will submit a software project during the week from 2 to 6 of December of the course. This submission must include the UML diagrams of the final design, the code and the documentation about the implementation details. The software must run correctly on the computers of the educational laboratories. In the assessment, the professor will consider both the correct execution of the program and the designed adopted in the development. With this test CE53 and CE50 competence will be assessed.	15 s
Practical tests, real task execution and / or simulated.	During the week from 9 to 13 of December of the educational period, the students will have an interview with the professor where they must answer questions related to the software project submitted (e.g. justify design decisions and propose solutions to face modifications in the developed functionalities. The two students of each group must be present in this interview. The questions must be answered individually with the goal of checking the authorship, the degree of understanding and implication of the student in the development of the project. If the student fails in proving these aspects, the student will have to take a practical exam in the computer room in the official date published in www.teleco.uvigo.es.	15 e
Case studies / analysis of situations	The students, organized into groups of 2 people, will submit and present in the computer room the design defined for the project software, including UML class diagrams. This design will be submitted during the week from 4 to 7 of November of the course. With this test CE51 and CE52 competences will be assessed.	10
Troubleshooting and or exercises	/Each student will take a final exam in the official date published in www.teleco.uvigo.es, which will consist of the following types of questions: resolution of problems, short-answer questions about the theoretical concepts explained in master sessions, true/false assessments, multiple-choice tests. Note that support materials are not allowed. The number and the combination of the aforementioned questions will be defined for each particular exam. With this test CE51 and CE53 competences will be assessed.	50
Practical tests, real task execution and / or simulated.	The students, organized into groups of 2 people, will submit the Java initiation practices proposed in the computer room. This submission will take place during the week from 21 to 25 of October of the course.	10

# Other comments on the Evaluation

There exist two mechanisms for the assessment of students in this subject: continuous assessment (CA) and traditional assessment (TA). Regardless of the considered assessment mechanism, the pass mark for the subject is 5 out of 10.

The students must choose one of the possible mechanisms by bearing in mind the following conditions:

- CA includes the 5 tests described above.
- In both cases, CA and TA, students must do a laboratory project. To easy the selection between CA and TA, students will have to see in Faitic Web the project to do since 20 September.
- In TA regimen the proyect must be made in an individual form.
- Students who sit CA must submit during the week from 4 to 7 of November of the course, the UML class diagrams of the proposed software project. By the submission of this practice the student makes a commitment to be assessed via CA, thus renouncing the TA mechanism. In virtue of this commitment, these students will not be listed as "Not Present".
- Students who do not submit the UML-compliant design during the week from 4 to 7 of November of the course, renounce to the CA, thus being assessed through the TA mechanism. Note that it will not be possible to join the CA in the next tests.
- CA tests will be only carried out in the dates defined by the teachers. These tests cannot be repeated out of these dates.
- The grades obtained in the CA and other exams and practical projects are only valid for the current academic year.
- CA will be just considered in January. In the rest of examination sessions (i.e., May, July and others) only TA will be valid.

The students who opt for CA will be assessed by the tests described above:

- Java initiation practices (10%). 2 people groups. It is the 5th test described in the CA assessment mechanism.
- Software project (40%). 2 people groups. It includes three parts: design(10%), implementation (15%) and interview (15%). They are the 3rd, 1st and 2nd tests, respectively.
- Exam about theoretical concepts and use cases. It is the 4th test described above.

The students who opt by the TA mechanism will be assessed as follows:

- An exam whose description matches the 4th CA test. The result of this exam will be 50% of the final remark of the student.
- A software project that must be submitted <u>individually</u> during the week from 2 to 5 of December of the course. The submission must include the class diagrams of UML-compliant design, and the documentation to explain the implementation details. The program must run correctly on the computer in the laboratory. The assessment of the project will consider both the right execution of the program and the proposed design. The weight of the project will be 30% of the final remark of the student.
- A personal interview where the student must answer the questions posed by the professor, with regard to main design decisions and the way to face possible modifications in the developed functionalities. This interview will take place during the week from 9 to 13 of December of the course in the computer room. The weight of this interview will be 20% of the final remark of the student.

Students who opt for the **examination sessions of July** cannot be assessed via CA, so that only TA is valid. These students will be assessed as per the following tests:

- A final exam (whose description matches the 4th CA test) that will take place in the official date published at www.teleco.uvigo.es. The result of this exam will be 50% of the final remark of the student.
- A practical exam in the computer room, which will take place in the official date published at www.teleco.uvigo.es, where the student must develop a Java program as per the guidelines described by the professor. The weight of this practical exam in the final remark will be 50%.

## Sources of information

Basic references:

[2] [Introduction to Java programming]. Y. Daniel Liang, 8ª edición. 2010, Pearson.

Other references:

[1] [Programación orientada a objetos con Java: una introducción práctica usando BlueJ]. D. J. Barnes, M. Kölling. 3ª edición. 2007, Pearson.

[3] [Data Scructures & Algorithms in Java]. Michale T. Goodrich, Roberto Tamassia, 5ª edición. 2010, Willey.

[4] [Java Tools]. Andreas Eberhart, Stefan Fischer. 2002, Wiley

[5] []*ava In A Nutshell*]. David Flanagan, 5ª edición. 2005, O'Reilly.

[6] []*Thinking in Java*[]. Bruce Eckel, 4ª edición. 2006, Prentice Hall

[7] [Learning Java]. Patrick Niemeyer, 3ª edición. O'Reilly Media

[8] [How to Think Like a Computer Scientist. JavaTM Version]. 4<sup>a</sup> version. Online: http://www.greenteapress.com/thinkapjava/

[9] [Java notes]. Fred Swartz. Online: http://www.leepoint.net/notes-java/index.html

[10] [Java SE. Oracle]. Online: http://www.oracle.com/technetwork/java/javase/downloads/index.html

[11] [Java 2 Platform Standard Edition 5.0. API Specification]. Online: http://download.oracle.com/javase/1.5.0/docs/api/

[12] []The Java Tutorials[]. Oracle. Online: http://download.oracle.com/javase/tutorial/

[14] []*Open-oriented Analysis and Design with Applications*[]. Grady Booch, Robert Maksimchuk, Michael Engel, Bobbi Young, Jim Conallen, Kelli Houston, 3ª edición. 2007, Addison Wesley.

[17] [[Fundamentals of Object-oriented design in UML]]. Meilir Page-Jones. 2002, Addison Wesley.

#### Recommendations

### Subjects that it is recommended to have taken before

(\*)Programación I/V05G300V01205