UNIVERSIDAD DE LA RIOJA

OBJECT ORIENTED PROGRAMMING
2016-2017

Bachelor Degree:	Software Engineering		801G
Course title:	Object oriented programming		828
Year/Semester:	2/1S	ECTS Credits:	6

DEPARTMENT

Address:	CCT Building, C/ Madre de Dios 53				
City:	Logroño	Province:	La Rioja	Postal code:	26007
Phone number:	941299452	Email address:			

ENGLISH-FRIENDLY FACULTY

Name:	Jesús María Aransay Azofra		
Phone number:	941299438	Email address:	jesus-maria.aransay@unirioja.es
Office:	3245	Building:	CCT

Name:	Jónathan Heras Vicente		
Phone number:	941299673	Email address:	jonathan.heras@unirioja.es
Office:	3232	Building:	CCT

CONTENTS

1. Objects and classes in Object Oriented programming
    1.1 Representing information by means of objects
    1.2 Attributes or state
    1.3 Methods or behavior
    1.4 Abstracting objects into classes
    1.5 The need and relevance of class constructors: default constructor, user defined constructors
    1.6 Access methods and how to modify the state of an object
    1.7 Static or class attributes and methods
    1.8 Access modifiers: the need for public and private modifiers
    1.9 Information hiding: different ways to represent an object without modifying its behavior
    1.10 Introduction to UML: representation of objects and classes
    1.11 C++: class declaration and object construction
    1.12 Java: classes declaration
2. Relations among classes. Class inheritance
    2.1 Communication among classes
    2.2 Classes containing objects as attributes: some well-known examples
    2.3 Specialisation/generalisation relationships
    2.4 Definition of class inheritance
    2.5 Advantages of using inheritance: code reuse and polymorphism
    2.6 Redefinition of methods in subclasses
    2.7 Access modifier “protected”: use cases
    2.8 Representing class inheritance in UML
    2.9 Programming in Java and C++ inheritance relationships
3. Using and defining polymorphic methods
    3.1 Definition of polymorphism and use advantages

3.2 Obtaining polymorphic methods in C++: dynamic memory allocation and virtual methods
3.3 Polymorphism in Java
3.4 Polymorphic methods in some previous case studies

4. Abstract classes and interfaces
4.1 Definition of abstract methods in OOP. Some use cases
4.2 Relationship between polymorphism and abstract methods
4.3 Definition and advantages of fully abstract classes or interfaces
4.4 UML representation of abstract methods, abstract classes and interfaces
4.5 C++ implementation of abstract methods and abstract classes
4.6 Java implementation of abstract methods and interfaces

5. Exceptions in Java
5.1 Definition of exceptions in programming
5.2 Different types of exceptions / errors and how to treat them
5.3 Dealing with exceptions: declaration, construction, throwing and handling
5.4 Programming with exceptions in Java. Using API exceptions and own defined exceptions

REFERENCES

Title
Thinking in Java; Bruce Eckel
Thinking in C++: Bruce Eckel
Objects first with Java : a practical introduction using BlueJ
Java Generics and Collections; Maurice Naftalin and Philip Wadler
Effective Java; Joshua Bloch
The unified modelling language reference manual; James Rumbaugh, Ivar Jacobson, Grady Booch Design patterns: elements of reusable object-oriented software; Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
Objects, abstraction, data structures and design using Java; Elliot B. Koffman, Paul A.T. Wolfgang
Java in a nutshell; David Flanagan
Practical object-oriented development with UML and Java; Richard C. Lee, William M. Tepfenhart
The C++ programming language; Bjarne Stroustrup
The Java Language Specification; James Gosling, Bill Joy, Guy Steele, Gilad Bracha
Java Examples in a Nutshell; David Flanagan
Effective C++

EVALUATION SYSTEM

Final exam: 70% (This part of the grade can be recovered)

Laboratory work and weekly reports: 30% (This part of the grade cannot be recovered)

The students must obtain more than 50% of the maximum possible qualification both in the "Final exam" and in the "Laboratory work and weekly reports". The "Laboratory work and weekly reports" also consider student attendance to both theoretical classes and laboratories.