

230086 - POO - Object Oriented Programming

Coordinating unit:	230 - ETSETB - Barcelona School of Telecommunications Engineering		
Teaching unit:	701 - AC - Department of Computer Architecture		
Academic year:	2019		
Degree:	BACHELOR'S DEGREE IN TELECOMMUNICATIONS TECHNOLOGIES AND SERVICES ENGINEERING (Syllabus 2015). (Teaching unit Compulsory)		
ECTS credits:	6	Teaching languages:	Catalan, Spanish

Teaching staff

Coordinator:	Cruellas Ibarz, Juan Carlos
Others:	Abadal Cavalle, Sergi Gil Gómez, María Luisa Guerrero Zapata, Manel Macías Lloret, Mario Perello Muntan, Jordi Rodríguez Luna, Eva

Degree competences to which the subject contributes

Transversal:

06 URI N1. EFFECTIVE USE OF INFORMATION RESOURCES - Level 1. Identifying information needs. Using collections, premises and services that are available for designing and executing simple searches that are suited to the topic.

Learning objectives of the subject

To understand what the object-oriented paradigm is. To understand its most remarkable concepts: class, object, encapsulation, aggregation, inheritance and polymorphism.

To know how to apply sort and search algorithms studied in 1A term in Java language, and study additional algorithms. Study algorithms for advanced structures (graphs).

To know how to generate a Java program from a design given by UML class diagrams and the specification of the methods of the involved classes.

Study load

Total learning time: 150h	Hours large group:	39h	26.00%
	Hours small group:	26h	17.33%
	Self study:	85h	56.67%

230086 - POO - Object Oriented Programming

Content

<p>Lesson 1. The paradigm of object orientation: classes and objects</p>	<p>Learning time: 3h 30m Theory classes: 1h 30m Self study : 2h</p>
<p>Description: Presentation of the paradigm of object orientation. Object and class concepts. Attributes and methods.</p> <p>Specific objectives: Formalize the concept of an object as an entity that gathers data (attributes) and functions (methods). Formalize the concept of class.</p>	
<p>Lesson 2. Classes and Objects in Java</p>	<p>Learning time: 21h Theory classes: 6h Self study : 15h</p>
<p>Description: Syntax class definition in Java. Defining attributes. Access modifiers: public, private. Definition of methods. Special methods: constructors, methods for accessing attributes (set and get). Creating Objects: new operator. Dynamic Memory Management Attributes and static methods. Its purpose and use. Constant and final modifiers.</p>	
<p>Lesson 3. Primitive types in Java and relevant classes</p>	<p>Learning time: 7h Theory classes: 2h Self study : 5h</p>
<p>Description: Java primitive types: integer types, real types, type character (difference from C), boolean, byte. Classes corresponding to the primitive types: Integer, Double, Float, Character, Boolean, Byte. String Class. Concept and relevant methods. From numerical values to their textual representation as Strings and vice versa.</p>	

230086 - POO - Object Oriented Programming

<p>Lesson 4. Containers</p>	<p>Learning time: 28h Theory classes: 8h Self study : 20h</p>
<p>Description: Concept of container object. Container types: sequences, dictionaries and sets. Implementation of sequences in Java: ArrayList and LinkedList. Most relevant methods. Iterators. Concept and its role in the path sequence. Implementation of dictionaries in Java: HashMap. Most relevant methods. Getting partial views of a dictionary: methods keySet (), values ??() Implementation of sets in Java: HashSet. Most relevant methods. Identification of the type of container that should be used in each case.</p>	
<p>(ENG) Tema 5. Introduction to UML. Class diagram</p>	<p>Learning time: 11h Theory classes: 3h Self study : 8h</p>
<p>Description: The UML as a language of graphical representation of a program. The class diagram. Relations between classes, associations and dependencies. Complete definition of an association name, directionality, cardinality. Implementation of associations in Java code.</p>	
<p>Lesson 6. Inheritance and polimorfism</p>	<p>Learning time: 24h Theory classes: 8h Self study : 16h</p>
<p>Description: Concept of inheritance. Superclasses and derived classes (subclasses). Inheritance of attributes and methods. Constructors in subclasses. The inheritance as a relationship between classes. Representation of inheritance in class diagrams. The object instance of a subclass are also objects instance of the superclass. The class Object class as the root class in classes hierarchy of Java. The toString () method of Object. Concept of polymorphism. Polymorphic methods. The Interface in Java. The interface as simulator of multiple inheritance. The interface as a mechanism that allows one object to present various types of behavior. Sorting and searching in polimorphic arrays.</p>	

230086 - POO - Object Oriented Programming

<p>Lesson 7. Exceptions</p>	<p>Learning time: 11h 30m Theory classes: 4h 30m Self study : 7h</p>
<p>Description: content english</p> <p>Specific objectives: The concept of exception. Creating, throwing and catching Java exceptions. Exception handling.</p>	
<p>Lesson 8. Input/Output in Java</p>	<p>Learning time: 4h Theory classes: 1h 30m Self study : 2h 30m</p>
<p>Description: Concept of stream. Hierarchy of Java classes for Input/Output based on streams. Classes for managing Input/Output of characters to/from files.</p>	
<p>Lesson 9. Graphs and algorithms for their manipulation</p>	<p>Learning time: 12h Theory classes: 4h Self study : 8h</p>
<p>Description: Main relevant concepts of graphs. Most relevant algorithms for managing graphs.</p> <p>Related activities: Problems solving and one lab session.</p> <p>Specific objectives: Concept of graph Elements: vertex and edges. Study and consolidate the following elements below: Graph's representation: adjacency list, and adjacency matrix. Java implementation of a graph. Traversal algorithms: "breadth-first search" and "depth-first search". Finding paths in a graph: "single-source shortest path problem" and "all-pairs shortest path problem"</p>	

230086 - POO - Object Oriented Programming

Planning of activities

TEST	Hours: 2h Theory classes: 2h
LAB SESSION 1: IDE, DEBUGGING AND BASIC ALGORITHMS	Hours: 2h Laboratory classes: 2h
Description: Introduction to NetBeans IDE. Creating projects. Editing code. Debugging. Basic algorithms implementation	
LAB SESSION 2: ARRAYS AND SORTING ALGORITHMS.	Hours: 2h Laboratory classes: 2h
Description: Usage of arrays in Java	
LAB SESSION 3: SEARCH ALGORITHMS	Hours: 2h Laboratory classes: 2h
Description: Using objects in Java. String and StringBuffer. Using relevant String methods: split (), indexOf (), etc.	
LAB SESSION 4: IMPLEMENTATION OF CLASSES AND USE OF OBJECTS IN JAVA	Hours: 2h Laboratory classes: 2h
Description: Defining and instantiating a class.	
LAB SESSION 5: ALGORITHMS FOR USING LISTS, DICTIONARIES AND SETS IN JAVA.	Hours: 2h Laboratory classes: 2h
Description: Using containers in Java.	
LAB SESSION 6: BUILDING A SMALL APPLICATION; THE MAZE (1)	Hours: 2h Laboratory classes: 2h
Description: First step in building a small application with containers, inheritance, polimorphism, exceptions, and input/output (1)	

230086 - POO - Object Oriented Programming

LAB SESSION 7: INHERITANCE AND POLYMORPHISM, THE MAZE (2)	Hours: 2h Laboratory classes: 2h
Description: Second version of a small application incorporating inheritance and polimorphism (2)	
LAB SESSION 8: EXCEPTIONS AND INPUT/OUTPUT, THE MAZE (3)	Hours: 2h Laboratory classes: 2h
Description: Implementation of a small application incorporating exceptions, and input/output (3)	
LAB SESSION 9: ALGORITHMS FOR GRAPHS, THE MAZE (4)	Hours: 2h Laboratory classes: 2h
FINAL EXAM	Hours: 3h Laboratory classes: 3h
LAB SESSION: PROJECT WORKING SESSION 1	Hours: 2h Laboratory classes: 2h
Description: Working session on deliverable 1 of the project	
LAB SESSION: PROJECT WORKING SESSION 2	Hours: 2h Laboratory classes: 2h
Description: Working session on deliverable 2 of the project	

Qualification system

- Lab project and lab test 20% to 25%
- Short exam 20% to 25%
- Final exam 55%

Regulations for carrying out activities

In order to pass, it is mandatory to deliver the lab project and to do the lab test

230086 - POO - Object Oriented Programming

Bibliography

Basic:

Eckel, Bruce. Piensa en Java. 4a ed. Madrid: Prentice Hall, 2007. ISBN 9788489660342.

García de Jalón de la Fuente, Javier [et al.]. Aprende Java como si estuviera en primero [on line]. San Sebastián: Universidad de Navarra. Escuela Superior de Ingenieros Industriales, 1999 [Consultation: 08/07/2015]. Available on: <<http://www1.ceit.es/asignaturas/Informat1/ayudainf/aprendainf/Java/Java2.pdf>>.

Complementary:

Documentació d'API de Java [on line]. [Consultation: 16/07/2015]. Available on: <<https://docs.oracle.com/javase/8/docs/api/>>.

Tutorías de Java de Oracle [on line]. [Consultation: 16/07/2015]. Available on: <<https://docs.oracle.com/javase/tutorial/>>.

Others resources:

Computer material

Java Development Kit

Java Development Kit

<https://netbeans.org/>

NetBeans