



## Course Syllabus

<b>Course Code</b>	<b>Course Title</b>	<b>ECTS Credits</b>
COMP-212	Object-Oriented Programming	6
<b>Prerequisites</b>	<b>Department</b>	<b>Semester</b>
COMP-113	Computer Science	Fall
<b>Type of Course</b>	<b>Field</b>	<b>Language of Instruction</b>
Required	Computer Science	English
<b>Level of Course</b>	<b>Lecturer(s)</b>	<b>Year of Study</b>
1 <sup>st</sup> Cycle	Dr. Constandinos Mavromoustakis	2 <sup>nd</sup>
<b>Mode of Delivery</b>	<b>Work Placement</b>	<b>Corequisites</b>
Face-to-face	N/A	None

### Course Objectives:

The main objectives of the course are to:

- learn the basic principles of the object-oriented programming with specific reference to the Java programming language.
- introduce and acquire the knowledge for simple object-oriented concepts and for more complex (private classes, objects, encapsulation, inheritance and polymorphism).
- Understand the key Object Oriented Concepts (OO Concepts) required to build an OO system.
- learn different Object Oriented Analysis and Design approaches (OOAD) to architect and build object oriented systems.
- develop a way for efficient algorithmic thinking and problem solving using the object-oriented paradigm with the UML (Unified Modeling Language).
- learn the Java graphical user interfaces (GUI) and the associated libraries of SDK/Sun; describe, plan, and build simple applications using the concepts of object-oriented programming in the Java context. Student will be enforced to adopt object-oriented methods to a variety of problems, with emphasis on the Reverse Engineering (RE) paradigm.

## Learning Outcomes:

After completion of the course students are expected to be able to:

1. Analyze problems and find abstract OO solutions.
2. Understand basic principles of object-oriented program design/ advanced issues related to writing classes and methods-such as data, visibility, scope, method parameters, object references, and nested classes.
3. Exploit object-oriented principles and advanced java language features in the design and implementation of object-oriented programs/ understand the basic ideas behind class hierarchies, polymorphism, and programming to interfaces.
4. Explain the capabilities of several java API's and be able to use them appropriately.
5. Understand the basic programming concepts and problem solving techniques.
6. Understand the object-oriented, windows-based and event driven programming paradigms.
7. Produce good Object Oriented designs.
8. Implement, test, maintain and refactor small to medium sized applications in Java/ develop API applications consisting of multiple source files.
9. Design write and execute programs in Java.
10. Have an understanding of the basic concepts of object oriented programming.
11. Have a knowledge of the core language of Java-develop (write/debug/correct) working Java source code and programs.
12. Learn and use the different views of the UML-learn the reverse engineering procedures using the UML model paradigm.

## Course Content:

1. Overview of programming languages/History of Java-Intro to Java applications/Intro to classes and objects/Classes, objects, methods, and variables.
2. Declaring classes and instantiating objects/Constructors/Garbage Collection/finalize.
3. Inheritance/Interfaces and Abstract Classes and methods/Creating templates and packages.
4. Methods/Program modules in Java/ Java API packages /Random numbers/ Method overloading.
5. Classes and Objects: A Deeper Look/ Controlling Access to methods/Constructors/Final instance variables/Creating packages/Encapsulation.
6. Event-Driven Programming/Exception Handling, IO/Exceptions and Assertions.
7. GUI Components/ Simple GUI-Based Input/Output with JOptionPane/ Overview of Swing Components.
8. Polymorphism/Java API and Interfaces.
9. Exception handling/JException Hierarchy/Declaring New Exception Types.

10. Introduction to Java Applets/Applets and HTML/Applet Life-Cycle Methods/Examples
11. ArrayList<E> for expandable arrays/ Collections Vector class including Vector, Stack, and Hashtable as Object class references-Files and Streams/ Class File/ Sequential-Access Text Files-Networking issues using java/Sockets/.
12. Object-oriented paradigm with the UML (Unified Modelling Language).

**Learning Activities and Teaching Methods:**

Lectures, Lab Presentations, Lab Tutorials, Practical Exercises and Assignments.

**Assessment Methods:**

Tests/Quizzes, Design project, Homework, Project, Mid-Term, Final Exam.

**Required Textbooks / Readings:**

Title	Author(s)	Publisher	Year	ISBN
Java™ How to Program, 9 <sup>th</sup> Edition	Harvey M. Deitel, Paul J. Deitel	Prentice Hall Inc. 2011	2016	ISBN-10: 0132575663  ISBN-13: 9780132575669  pp. 1536

**Recommended Textbooks / Readings:**

Title	Author(s)	Publisher	Year	ISBN
Thinking in Java, 2nd Ed.	Bruce Eckel	Prentice Hall Inc.	2009	9780131872486
Introduction to Java Programming, 8th Edition	Daniel Liang	Prentice Hall Inc.	2010	10: 0136042589