University of Nicosia, Cyprus

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

Objectives of the Course:

The main objectives of the course are to:

- thoroughly discuss and acquire the knowledge and programming experience of basic principles of the object-oriented programming with specific reference to the Java programming language
- demonstrate and analyze the basic object-oriented concepts for simple concepts as well as for more complex (private classes, objects, encapsulation, inheritance and polymorphism)
- identify the key Object Oriented Concepts (OO Concepts) required to build an OO system
- critically assess different Object Oriented Analysis and Design approaches (OOAD) to architect and build object oriented systems
- demonstrate and analyze a way for efficient algorithmic thinking and problem solving using the object-oriented paradigm with the UML (Unified Modelling Language)
- design, practice and develop using the Java graphical user interfaces (GUI) applications with the associated API libraries of SDK/Oracle
- critically assess, plan, and build simple applications using the concepts of object-oriented programming in the Java context
- introduce state-of-the art application development in the area of object-oriented implementation methodologies 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. identify basic principles of object-oriented program design/ advanced issues related to extrapolate manipulation of 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
4. identify the basic ideas behind class hierarchies, polymorphism, and programming to interfaces
5. explain the capabilities of several java API's and demonstrate appropriately the utilization of them
6. identify the basic programming concepts and problem solving techniques
7. identify the object-oriented, windows-based and event driven programming paradigms
8. formulate and organize an excellent Object Oriented design skills
9. implement, test, maintain and refactor small to medium sized applications in Java/ develop API applications consisting of multiple source files
10. design write and execute programs in Java.
11. demonstrate and analyze the basic concepts of object oriented programming
12. critically assess the abstractions of the Object Oriented design core language of Java
13. design and develop (write/debug/correct) Java source code and GUI programs with specified requirements
14. establish a solid knowledge and utilize the different views of the UML
15. research in state-of-the art areas for the up-to-date reverse engineering procedures using the UML model paradigm

Course Contents:

1. Overview of programming languages/History of Java
2. Intro to Java applications/Intro to classes and objects/Classes, objects, methods, and variables
3. Declaring classes and instantiating objects/Constructors
4. Inheritance/Interfaces and Abstract Classes and methods/Creating templates and packages
5. Decisions/Algorithms, pseudocode, control structures
6. Methods/Program modules in Java/ Java API packages /Random numbers/ Method overloading
7. Classes and Objects: A Deeper Look/ Controlling Access to methods/Constructors/Final instance variables/Creating packages/Encapsulation
8. Event-Driven Programming/Exception Handling, IO/Exceptions and Assertions
9. GUI Components/ Simple GUI-Based Input/Output with JOptionPane/ Overview of Swing Components
10. Polymorphism/Java API and Interfaces
11. Exception handling/JException Hierarchy/Declaring New Exception Types
12. Java Applets
13. Applets and HTML/Applet Life-Cycle Methods/Examples
14. Sorting and searching/ Searching arrays/Sorting arrays
15. Collections such as ArrayList<>, Vector, Stack, and Hashtable as Object class references
16. Files and Streams/ Class File/ Sequential-Access Text Files

17. Networking issues using java/Sockets/
18. Client/Server Interaction with Stream Socket Connections/Serialization
19. Object-oriented paradigm with the UML (Unified Modelling Language)
20. An Overview of Java Security-Java enabled Network Security Framework

Learning Activities and Teaching Methods:

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

Assessment Methods:

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

Required Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Harvey M. Deitel, Paul J. Deitel,	Java™ How to Program, 8th Ed.	Prentice Hall Inc.	2010	0136053068

Recommended Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Bruce Eckel, ,	Thinking in Java, 2nd Ed.	Prentice Hall Inc.	2004.	9780131872486
Y. Daniel Liang, ,	Introduction to Java Programming, 7th Edition	Prentice Hall Inc.	2008	10: 0136042589