Course Title	Object Oriented Programming				
Course Code	ACSC382				
Course Type	BSc Computer Science: Required Course BSc Computer Engineering: Computer Elective				
Level	BSc (Level 1)				
Year / Semester	2nd year / 4th semester				
Teacher's Name	Dr. Achilleas Achilleos				
ECTS	5	Lectures / week	2	Laboratories/week	2
Course Purpose	The aim of this course is to help students understand the basic concepts of object-oriented programming and enable them to critically think on how to apply these concepts in practice to solve real world problems. The course introduces and applies in practice key concepts such as object-orientation, classes, objects and instantiation, modularity and reusable components, abstraction, encapsulation, inheritance, polymorphism, I/O and Serialization, Graphical User Interfaces (GUI) programming, event and exception handling.				
Learning Outcomes	Upon successful completion of the course students will be able to: • Explain the transition from data types in procedural programming paradigm to abstract data types in object-oriented programming. • Compare and describe the shift from simple variables to classes and objects, and from libraries of functions to packages of classes. • Describe and explain the principles of object orientation, including abstraction, data encapsulation and information hiding, message passing, inheritance and polymorphism. • Use and apply these principles through their implementation in the Java programming language. • Write programs applying the principles of object orientation in other object-oriented programming languages, such as C#. • Construct complex programs using built-in classes and packages in Java programming environment to create reusable and extensible code based on object-oriented concepts. • Implement high quality programs in Java using exception handling mechanism, streams and object serialization. • Explain and outline the basic concepts such as layout managers, components and Java libraries for Graphical User Interface (GUI) programming. • Design and implement GUI-based software applications in Java that follow the principles of object-oriented programming and event-based programming.				

Prerequisites	**ACSC183.**	Corequisites	**None.**
Course Content	**1. Object Oriented Programming (OOP) and the Java Language (2 Weeks)** - Hardware and Software. Information Binary Representation and Memory. Generations of Programming Languages. The Software Development Process. Object Oriented Programming Basics. Why Java? Java Virtual Machine and Byte Code. Classes, Objects, Messages and Methods. User Interfacing Methods. JDK installation and use. IDEs and Editors – Edit, Compile and Execute in Java. Compile Time Errors. **2. Java Syntax, Semantics and Execution Control (2 Weeks)** - Language elements. Java data types, literals, statements, reserved words, variables, type casting. Classes and Objects: Abstraction. Java Methods. Objects communication. Class Definition, Instances and State. Constructors. Reading User Input: Java IO objects, streams and serialization. Programming Errors and Debugging. Control Structures: Sequence, Selection, Repetition, Recursion. Conditions. Compound Boolean Expressions. **3. Reuse, Extensibility, Reliability and Collaboration. (3 Weeks)** - Packages: Modularity and Reusable components. Namespace. Standard Packages, Classes and Methods available in Java. Autoboxing and Unboxing. Modules. Class Members and Modifiers. Data and method access types. Classes and Objects – Instantiation. Classes, Objects, and Memory. The Structure and Behaviour of Methods. Overloading Methods. Static Modifier: Class vs Instance Methods and Variables. **4. Advanced Object-Oriented Programming Concepts (1 Week)** - Inheritance: Super class and generalization relationship. Abstraction: Definition, Abstract class, Interface. Polymorphism: Extensibility by building class hierarchy. Encapsulation: Information hiding. **5. Arrays, Manipulation of Arrays and Operations on Arrays (3 Weeks)** - Arrays Conceptual Overview. Declaring Arrays. Simple Array Manipulations. Looping through Arrays. Parallel Arrays. Two-Dimensional Arrays. Enhanced for loop in Java. Arrays and Methods. Linear Search. Binary Search. Comparable Interface. Sorting: Selection Sort, Bubble Sort. Arrays of Objects: The Class java.util.ArrayList. **6. Graphical User Interface Programming and Event Handling (2 Weeks)** - Graphical User Interface (GUI): Definition. Java GUI Objects. The java.awt and javax.swing packages for graphical user interface programming. Event Handling. Event Source Objects. Event Listener Objects. Event		

	Types. The ActionListener Interface. Layout Managers: FlowLayout, BorderLayout, GridLayout.
Teaching Methodology	The methodology used to conduct the course is structured around lectures and laboratory examples/exercises, so that students gain theoretical knowledge as well as practical skills. The taught part of course is delivered to the students with the help of computer presentations. Furthermore, the principles are demonstrated by means of specific examples that help students engage with solving problems programmatically. Lecture notes in the form of presentations and code are available through the e-learning system for students to use in combination with the recommended textbooks and references. Lectures are supplemented with computer laboratories that include rigorous demonstrations of taught concepts and experimentation with the Java programming language. Additionally, during laboratory sessions, students apply their gained knowledge and identify the principles taught in the lectures by means of working on different tasks and problems. Students are also allocated exercises during the laboratories to improve both their individual skills and teamwork. Eight of these exercises are submitted at the end of the laboratory sessions. Moreover, two assignments or a Java software project is assigned to students to test their understanding of the basic principles of object-oriented programming. Finally, the course assessment is completed through a three-hours final exam at the end of the semester.
Bibliography	**Textbooks:** 1. Wu, C. Thomas, An Introduction to Object-Oriented Programming with Java, 5th edition, McGraw-Hill, 2010. Paperback: 1008 pages, Publisher: McGraw-Hill Education; 5th edition (March 24, 2009), Language: English, ISBN: 0073523305. **References:** 1. Bruce Eckel, Thinking in Java (4th Edition), Published by: PRENTICE HALL (NEW JERSEY), 2008, ISBN:9780131872486. 2. Joshua Bloch, Effective Java, 3rd Edition, Publisher: Addison-Wesley Professional, Release Date: December 2017, ISBN: 9780134686097. 3. Oracle Reference Guide – https://docs.oracle.com/javase/tutorial/ 4. W3Schools Java Programming – https://www.w3schools.com/java/
Assessment	• Eight Laboratory exercises for a total of: 20% • Two Assignments or Java Project for a total of: 20% • A three-hours Final Exam: 60%
Language	English.