



East West University
Department of Computer Science and Engineering
Course Outline of CSE110

Course Information

Course: CSE110 Object Oriented Programming

Credit and Teaching Scheme:

	Theory	Laboratory	Total
Credits	3.0	1.5	4.5
Contact Hours	3 Hours/Week for 13 Weeks	3 Hours/Week for 13 Weeks	6 Hours/Week for 13 Weeks

Prerequisite: CSE103 Structured Programming

Instructor Information

Instructor: Dr. Mohammad Salah Uddin
Associate Professor, Department of Computer Science and Engineering
Office: Room # 648
Tel. No.: 09666775577 (hunting) ext. 395
E-mail: uddin@ewubd.edu
Course Repository: Google Classroom/(www.banglarbot.com/oop)
GTA:

Class and Office Hour:

Day	08:30-10:00	10:10-11:40	11:50-01:20	01:30-03:00	03:10-04:40	04:50-06:20
Sunday			Office Hour	CSE407 (5) AB3-802	CSE227 (1) LAB Room#215	CSE227 (1) LAB Room#215
Monday				Office Hour	CSE106 (4) Room#221	
Tuesday			CSE227 (1) Room#FUB-301	Office Hour	CSE110 (3) Room#221	
Wednesday		CSE110 (3) LAB Room#227	CSE110 (3) LAB Room#227	Office Hour	CSE106 (4) Room#221	
Thursday		Office Hour	CSE227 (1) Room#FUB-301	CSE407 (5) AB3-802	CSE110 (3) Room#221	

Course Objective

This course presents a conceptual and practical introduction to object-oriented programming (OOP). The course will cover general principles of programming in object-oriented frameworks to enhance transferable skills, such as programming, designing, and problem-solving skills. This

course introduces object-oriented concepts and develops OOP programs which provides solutions to real-world object-oriented problems. Java is primarily chosen as the programming language in this course. Knowledge of this course will be needed as prerequisite knowledge for CSE207 Data Structures.

Course Outcomes (COs)

After completion of this course, students will be able to:

CO1	Understand and apply the fundamental principles of object-oriented programming in the target language to solve problems using an object-oriented approach.
CO2	Use the principles of inheritance, polymorphism, abstraction, and interfaces to develop reusable and modular code.
CO3	Apply exceptions, generics, multithreading, inner classes, and perform file I/O operations to develop robust and data-driven applications.
CO4	Choose appropriate tools, perform and demonstrate skills, and write a report to design, build, and test interactive applications that incorporate a graphical user interface (GUI) with the option of using multithreading and database connectivity in the target language to solve real-life complex problems.

Course Topics, Teaching-Learning Methods, and Assessment Scheme

Lecture-Wise Subtopics / Details	Course Topic	Teaching-Learning Method	CO	Mark of Cognitive Learning Levels		Mark of COs	Exam (Mark)
				C2	C3		
Lecture 1	Course Overview & Course structure, Target Language Fundamentals.	Lecture, Class Discussion, Discussion Outside Class with Instructor/Teaching Assistant					
Lecture 2	Principles of Object-Oriented Programming and Basics of Elementary		CO1	5	5	10	Mid Term Exam

	Programming in target language (conditional branching, looping, methods, and arrays)	Do						(25)
Lecture 3	Describe objects and classes, and use classes to model objects. Demonstrate how to define classes and create objects. Create objects using constructors. Access objects via object reference variables. Define a reference variable using a reference type. Access an object's data and methods using the object member access operator (.).	Do						
Lecture 4	Define data fields of reference types and assign default values for an object's data fields. Distinguish between object reference variables and primitive data type variables. Distinguish between instance and static variables and methods. Define private data fields with appropriate getter and setter methods.							

	Encapsulate data fields to make classes easy to maintain.						
Lecture 5	Develop methods with object arguments and differentiate between primitive-type arguments and object-type arguments. Store and process objects in arrays. Create immutable objects from immutable classes to protect the contents of objects. Determine the scope of variables in the context of a class. Use the keyword this to refer to the calling object itself. [Class test 1 will be conducted on topics under CO1]						
Lecture 6	Define a subclass from a superclass through inheritance. Invoke the superclass's constructors and methods using the super keyword. Override instance methods in the subclass. Distinguish differences between overriding and overloading.	Do	CO2		10	10	

Lecture 7	Explore the <code>toString()</code> method in the <code>Object</code> class. Discover polymorphism and dynamic binding. Describe casting and explain why explicit downcasting is necessary.						
Lecture 8	Enable data and methods in a superclass accessible from subclasses using the <code>protected</code> visibility modifier. Prevent class extending and method overriding using the <code>final</code> modifier.						
Lecture 9	Design and use abstract classes.	Do					
Lecture 10	Specify common behavior for objects using interfaces. Define interfaces and define classes that implement interfaces.						
Lecture 11	Define a natural order using the <code>Comparable</code> interface. Make objects cloneable using the <code>Cloneable</code> interface. Explore the similarities and differences among concrete classes,					5	5

	abstract classes, and interfaces. [Class Test 02 will be conducted on topics under CO2]						
Lecture 12	Get an overview of exceptions and exception handling. Explore the advantages of using exception handling. Distinguish exception types: Error (fatal) vs. Exception (nonfatal) and checked vs. unchecked. Declare exceptions in a method header. Throw exceptions in a method. Write a try-catch block to handle exceptions.	Do	CO3		5	5	Final Exam (30)
Lecture 13	Explain how an exception is propagated. Obtain information from an exception object. Use the finally clause in a try-catch block. Use exceptions only for unexpected errors. Create chained exceptions.						
Lecture 14	Discover file/directory				5	5	

	properties, to delete and rename files/directories, and to create directories using the File class. Write data to a file using the PrintWriter class. Use try-with-resources to ensure that the resources are closed automatically. Read data from a file using the Scanner class. Understand how data is read using a Scanner. Develop a program that replaces text in a file.						
Lecture 15	Describe the benefits of generics. Use generic classes and interfaces. Define generic classes and interfaces.	Do					
Lecture 16	Explain why generic types can improve reliability and readability. Define and use generic methods and bounded generic types. Use raw types for backward compatibility.				5	5	

Lecture 17	Explain why wildcard generic types are necessary. Describe generic type erasure and list certain restrictions and limitations on generic types caused by type erasure.						
Lecture 18	Get an overview of multithreading. Develop task classes by implementing the Runnable interface. Create threads to run tasks using the Thread class.						
Lecture 19	Control threads using the methods in the Thread class. Run the code in the application thread. Execute tasks in a thread pool.				10	10	
Lecture 20	Use synchronized methods or blocks to synchronize threads to avoid race conditions. Synchronize threads using locks.						
Lecture 21	Nested Class: Inner Class, Static nested class.				5	5	
Lecture 22	Inner class and Nested static class examples, Shadowing.						

Lecture 23	Functional Interface, Lambda Expressions and its uses in different scenarios. [Class Test 03 will be conducted on topics under CO3]						
------------	---	--	--	--	--	--	--

Lab Exercises

Experiment	Teaching-Learning Method	CO	Marks of Cognitive Level	Mark of Psychomotor Level		Mark of Affective Level	Mark of COs
			C3	P2	P3	A2	
Java Basics of Elementary Programming, Conditional Statements	Lab Experiment and Result Analysis and Discussion with Instructor, Post-Lab Report	CO4					
Looping, Nested Looping, Arrays Java Methods and library functions	Do	CO4					
Designing and Implementing simple Classes and Objects, Arrays of Objects etc.	Do	CO4					
Implementing associations of Classes, and Inner class	Do	CO4					
Designing and Implementing Inheritance and Polymorphism	Do	CO4					
Designing and Implementing Abstract Class and Interfaces	Do	CO4					

Understanding and Implementing Exceptions and File management, Collections, Generics	Do	CO4					
GUI using JavaFX	Do	CO4					
Lab Experiments		CO4	3	2.5	2.5	2	10
Lab Final (Exam)	Individual Exam	CO4	3	2.5	2.5	2	10
Total			6	5	5	4	20

Mini Project

Mini Project	Teaching-Learning Method	CO	Mark of Cognitive Levels		Mark of Psychomotor Levels		Mark of Affective Levels	Mark of COs
			C3	C4	P2	P3	A2	
Mini Project including Report and Presentation	Moderately complex Project with report writing, and oral/poster presentation	CO4	1	1	1	1	1	5

Overall Assessment Scheme

Assessment Area	COs				Assessment Area Mark
	CO1	CO2	CO3	CO4	
Class Participation & viva					5
Class Test/Quizzes	5	5	5		15
Mid Semester Assessment	10	15			25
Final Exam			30		30
Lab Performance				10	10
Lab Final				10	10
Mini Project Demonstration				5	5
Total Mark	15	20	35	25	100

Teaching Materials/Equipment

Text Book:

- Introduction to Java Programming by Daniel Liang
- Herbert Schildt, *Java: The Complete Reference*, 11th edition, McGraw-Hill Education (2023)

Reference Book:

- Paul Deitel, Harvey Deitel, Java™ How to Program Early Objects, 11th edition
- Walter Savitch, *Absolute Java*, Pearson (5th edition)
- Bert Bates and Kathy Sierra, *Head First Java*, O'Reilly Media (2nd edition)

Software/Tools:

- Java Development Kit (JDK 1.8)
<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- Any Integrated Development Environment (IDE) supporting Java preferably NetBeans or Eclipse
<https://netbeans.apache.org/download/index.html>, <https://www.eclipse.org/downloads/>
- Android Studio

Grading System

Marks (%)	Letter Grade	Grade Point	Marks (%)	Letter Grade	Grade Point
80-100	A+	4.00	55-59	B-	2.75
75-79	A	3.75	50-54	C+	2.5
70-74	A-	3.5	45-49	C	2.25
65-69	B+	3.25	40-44	C-	2
60-64	B	3.00	Below 40	F	0.00

Exam Dates

As per the schedule provided by the department and the university.

Academic Code of Conduct**Academic Integrity:**

Any form of cheating, plagiarism, personification, falsification of a document as well as any other form of dishonest behavior related to obtaining academic gain or the avoidance of evaluative exercises committed by a student is an academic offence under the Academic Code of Conduct and **may lead to severe penalties as decided by the Disciplinary Committee of the university.**

For details on Student Ethics and Academic Discipline, please visit <https://www.ewubd.edu/student-rules-regulation>

Special Instructions:

- Students are expected to attend all classes and examinations. A student **MUST** have at least 80% class attendance to sit for the final exam.
- Students will not be allowed to enter into the classroom after 10 minutes of the starting time.
- For plagiarism, the grade will automatically become zero for that exam/assignment.
- Normally there will be **NO make-up exam**. However, in case of **severe illness, death of any family member, any family emergency, or any humanitarian ground**, if a student miss any

exam, the student MUST get approval of makeup exam by written application to the Chairperson through the Course Instructor **within 48 hours** of the exam time. Proper supporting documents in favor of the reason of missing the exam have to be presented with the application.

- For **final exam**, there will be NO makeup exam. However, in case of **severe illness, death of any family member, any family emergency, or any humanitarian ground**, if a student miss the final exam, the student MUST get approval of **Incomplete Grade** by written application to the Chairperson through the Course Instructor **within 48 hours** of the final exam time. Proper supporting documents in favor of the reason of missing the final exam have to be presented with the application. **It is the responsibility of the student to arrange an Incomplete Exam within the deadline mentioned in the Academic Calendar in consultation with the Course Instructor.**
- All mobile phones MUST be turned to silent mode during class and exam period.
- There is **zero tolerance for cheating** in exam. Students caught with cheat sheets in their possession, whether used or not; writing on the palm of hand, back of calculators, chairs or nearby walls; copying from cheat sheets or other cheat sources; copying from other examinee, etc. would be treated as cheating in the exam hall. The only penalty for cheating is **expulsion for several semesters as decided by the Disciplinary Committee of the university.**