Course: CSE110 Object-Oriented Programming

Credits and Teaching Scheme

	Theory	Laboratory	Total
Credits	3	1.5	4.5
Contact Hours	3 Hours/Week for 13 Weeks + Final Exam in the 14th Week	3 Hours/Week for 13 Weeks	6 Hours/Week for 13 Weeks + Final Exam in the 14th Week

Prerequisite
CSE103 Structured Programming

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.

Knowledge Profile
K2: Conceptually-based mathematics, numerical analysis, statistics, and formal aspects of computer and information science

Learning Domains
Cognitive – C2: Understanding, C3: Applying
Psychomotor - P2: Manipulation, P3: Precision
Affective - A2: Responding

Program Outcomes (POs)
PO1: Engineering Knowledge

Complex Engineering Problem Solution
EP1: Depth of knowledge required
EP2: Range of conflicting requirements

Complex Engineering Activities
None

Course Outcomes (COs) with Mappings

After completion of this course students will be able to:

CO	CO Description	PO	Learning Domains	Knowledge Profile	Complex Engineering Problem Solving/ Engineering Activities
CO1	**Understand** and **apply** the basics of elementary programming in the target language and concepts related to the definition, creation and usage of classes and objects for writing object-oriented programs.	PO1	C2, C3	K2	-
CO2	**Understand** the advanced principles of OOP such as encapsulation, inheritance, polymorphism, abstract class and interface for implementing object-oriented programs.	PO1	C2	K2	-
CO3	**Apply** OOP concepts and constructs to design and implement complex object-oriented applications.	PO1	C3	K2	-
CO4	**Demonstrate** skills and **write** reports to design, build and test realistic, complex object-oriented application. **Use** appropriate language constructs; **perform** and **demonstrate** skills; and **write** report for developing programs for solving complex problems.	PO1	C3 P2, P3 A2	K2	-

Course Topics, Teaching-Learning Method, and Assessment Scheme

Course Topic	Teaching-Learning Method	CO	Mark of Cognitive Learning Levels		CO Mark	Exam (Mark)
			C2	C3		
Principles of Object-Oriented Programming and Basics of Elementary Programming in target language (conditional branching, looping, methods and arrays)	Lecture, Class Discussion, Discussion outside class with Instructor/TA	CO1	5	5	10	**Midterm Exam I (16)**
Introduction to Classes and Objects (Classes, Objects, Instance variables and instance methods, Constructors)	Do	CO1	6		6	
Inheritance and Polymorphism in OOP (super class, sub class, multiple-level inheritance, late binding)	Do	CO2	4		4	**Midterm Exam II (16)**
		CO3		4	4	
Abstract Class and Interfaces (differences, applicability and implementation)	Do	CO2	4		4	
		CO3		4	4	
Exception Handling in OOP and File handling using Text and Binary I/O	Do	CO3		10	10	**Final Exam (20)**
Implementation of Generics and GUI, Multi-threaded Programming, JDBC and other advanced topics	Do	CO3		10	10	

Laboratory Experiments and Assessment Scheme

Experiment	Teaching-Learning Method	CO	Marks of Cognitive Level	Mark of Psychomotor Level		Mark of Affective Level	CO Mark
			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	Do	CO4					
Java Methods and library functions	Do	CO4					
Designing and Implementing simple Classes and Objects, Arrays of Objects etc.	Do	CO4					
Implementing associations of Classes	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	Do	CO4					
Lab Exercises (Total)		CO4	10	2	1	2	15
Lab Exam	Individual Exam	CO4	5	1	1	0	7
Total			**15**	**3**	**2**	**2**	**22.0**

Mini Projects

Mini Project		CO	Mark of Cognitive	Mark of Psychomotor	Mark of Affective	CO Mark

	Teaching-Learning Method		CO4	Learning Level	Learning Levels		Learning Level	
				C3	P2	P3	A2	
Lab-based Mini Project including Report and Presentation	Group-based moderately complex digital circuit design project with report writing and oral/poster presentation	CO4		7	1	1	2	11

Overall Assessment Scheme

Assessment Area	CO				Other	PO Marks
	CO1	CO2	CO3	CO4		PO1
Class Participation					5	
Class Test/Quiz					10	
Midterm-I Exam	16	0	0	0		16
Midterm-II Exam	0	8	8	0		16
Final Exam	0	0	20	0		20
Laboratory Performance and Lab Exam	0	0	0	22		22
Mini Project	0	0	0	11		11
Total	**16**	**8**	**28**	**33**	**15**	**85**