Part A- Introduction

I. Course Code and Title : CSE 214 Object Oriented Programming Language Sessional

II. Credit : 1.5 Credit Hours

1. Rationale:

Object-oriented programming is a programming paradigm that includes or relies on the concept of objects, encapsulated data structures that have properties and functions and which interact with other objects

2. Course Description:

Encapsulation, classes and objects, access specifiers, static and non-static members; Constructors, Destructors and Copy Constructors; Array of objects, object pointers, and object references; In-line functions, friend functions, static functions; Inheritance: single and multiple inheritance; Polymorphism: overloading, abstract classes, virtual functions and overriding; Exception Handling; Object Oriented I/O ; Template functions and classes; Concept of Namespaces, Overview of Standard Template Library (Vectors & Iterators); Multi-threaded Programming, Abstract Data Type

3. Course Outcome (CO) Matrix:

No.	Course Learning Outcome (Upon completion of the course, the students will be able to)	Bloom's Taxonomy	CP	CA	KP	Assessment Methods
CO1	Practice object-oriented programming and design object-oriented solutions for small systems/problems involving multiple objects.	C6	1	3	5	E,O,ASG
CO2	Demonstrate good programming style and discuss the impact of style on developing and maintaining programs.	C3	1	-	1	O,R
CO3	Identify the relative merits of different algorithmic designs, programming constructs, and data structures.	P5	3	-	7	PR,Q,V
CO4	Write code, test, document and prepare a professional looking package for specified systems / problems	C3, C6	1,3	3	5	E, ASG
(CP- Complex Problems, CA-Complex Activities, KP-Knowledge Profile, T – Test ; PR – Project ; Q – Quiz; ASG – Assignment; Pr – Presentation; R - Report; V - Viva; F – Final Exam; MT – Mid Term)						

4. Mapping of COs and Program Outcomes (POs):

CLOs	Program Learning Outcomes (PLOs) (Appendix 1) H = High, M = Medium and L = Low										
	1	2	3	4	5	6	7	8	9	10	11	12
1									M			
2											M	
3								H				
4									M			

5. **Justification for CO-PO Mapping:**

Mapping	Level	Justification
CO1– PO9	High	To function effectively as an individual or a leader in a team, one must learn to design object-oriented solutions for small systems/problems involving multiple objects, which enhances teamwork and collaboration in software development.
CO2 – PO11	High	Recognizing the need for lifelong learning involves understanding and applying good programming practices. Demonstrating good programming style and discussing its impact on development and maintenance fosters continuous improvement and adaptability.
CO3 – PO8	High	Applying reasoning and taking responsibility in professional engineering practice require the ability to evaluate and choose the best algorithmic designs, programming constructs, and data structures for efficient solutions.
CO4 – PO9	Medium	To function effectively as an individual or team leader, one must develop skills in writing, testing, documenting, and preparing professional-quality software packages, ensuring clear communication and teamwork in software projects.

6. **Teaching Learning Strategy:**

Teaching and Learning Activities	Engagement (Hours)
Face-to-Face Learning ➢ Lecture ➢ Practical / Tutorial / Studio ➢ Student-Centered Learning	42
Self-Directed Learning	
➢ Non-face-to-face learning	
➢ Revision	
➢ Assessment Preparations	
Formal Assessment	
➢ Continuous Assessment	4
➢ Final Examination	3
	49

7. **Teaching Methodology:**

Lecture and Discussion, Co-operative and Collaborative Method, Problem Based Method

Part B- Content of the Course

8. Teaching Learning and Assessment Strategy Mapped with COs:

Class Time frame (Week)	Topics/Assignment	CLOs	Teaching/learning Strategy	Assessment Strategy
1	Introductory session on OOP, Transition from C to C++	CO2	Ref 1,2, Internet	Practice, Observation, Test, Assignment, Problem solving
2	Structure and Classes with namespace, Class and objects with access specifier	CO1	Ref 1,2, Internet	
3	Member Functions, In-line functions, Friend functions, Function Overloading	CO1, CO2	Ref 1,2, Internet	
4	Lab Test -1	CO1, CO2	Ref 1,2, Internet	
5	Introduction to the concept of Constructors and Destructors	CO1, CO2	Ref 1,2, Internet	Practice, Observation, Test, Assignment, Problem solving
6	Copy Constructors	CO1, CO2	Ref 1,2, Internet	
7	Inheritance: Introduction, derived and base classes, accessing base class members, access specified for 'protected'	CO1, CO2,CO3	Ref 1,2, Internet	
8	Multiple inheritance, Constructor and destructor in inheritance	CO1, CO2,CO3	Ref 1,2, Internet	
9	Virtual functions, runtime polymorphism and overriding Abstract class	CO1, CO2,CO3	Ref 1,2, Internet	
10	Operator overloading: Introduction, overloading of unary operators, binary operators, multiple overloading	CO1, CO2,CO3	Ref 1,2, Internet	
11	Template, File Handling	CO4	Ref 1,2, Internet	
12	Lab Test-2	CO1, CO2,CO3	Ref 1,2, Internet	
13	Viva + Quiz	CO1, CO2,CO3	Ref 1,2, Internet	Project Design and Implementation , Quiz, presentation, Viva
14	Project Presentation	CO4		

Part C- Assessment Strategy

Components		Grading	CO	Blooms Taxonomy
Continuous Assessment (60%)	Lab Test	30%	CO1	C2,C3
	Class Participation	10%	CO2	C1
	Assignment	10%	CO2	C6
	Report	10%	CO3	C1
Project (40%)		Presentation (10)	CO3	A1, A2
		Implementation (20)	CO4	S(P4)
		Viva (10)	CO3,CO4	A1, A2
Total Marks		100%		
(CO = Course Outcome, C = Cognitive Domain, P = Psychomotor Domain, A = Affective Domain)				

Part D- Learning Resources

9. Textbook:

1. Teach Yourself C++ by Herbert Schildt
2. Object Oriented Programming with C++ by E Balagurusamy
3. Complete Reference C++ by Herbert Schildt
4. Programming with C++ by Schaums Outline Series