



Course Title: Object Oriented Programming Using C++

Course Level: UG

Course Code: ES203

Credit Units: 04

L	T	P/ S	SW/ FW	TOTAL CREDIT UNITS
3	0	2	0	4

Course Objectives:

After finishing this course student will be able to:

- 1) get introduction of object oriented programming.
- 2) explore and implement the various features of OOP such as inheritance, polymorphism, Exceptional handling etc. using programming language C++.
- 3) to easily identify the basic difference between the programming approaches like procedural and object oriented

Pre-requisites:

Knowledge of Programming logic and techniques

Course Contents/Syllabus:

	Weightage (%)
Module I Introduction	
Descriptors/Topics	
<ul style="list-style-type: none">• Review of C• Difference between C and C++• Procedure Oriented and Object Oriented Approach	20

- Basic Concepts: Objects, classes, Principles like Abstraction, Encapsulation, Inheritance and Polymorphism. Dynamic Binding, Message Passing
- Characteristics of Object-Oriented Languages
- Introduction to Object-Oriented Modeling techniques (Object, Functional and Dynamic Modeling)

Module II Classes And Objects

Descriptors/Topics

- Abstract data types
- Object & classes, attributes, methods
- C++ class declaration
- Local Class and Global Class
- State identity and behaviour of an object
- Local Object and Global Object
- Scope resolution operator
- Friend Functions
- Inline functions
- Constructors and destructors, instantiation of objects
- Types of Constructors
- Static Class Data
- Array of Objects
- Constant member functions and Objects
- Memory management Operators

20

Module III Inheritance

Descriptors/Topics

- Inheritance
- Types of Inheritance
- access modes – public, private & protected
- Abstract Classes

<ul style="list-style-type: none"> • Ambiguity resolution using scope resolution operator and Virtual base class • Aggregation, composition vs classification hierarchies • Overriding inheritance methods, • Constructors in derived classes, <p>Nesting of Classes</p>	20
Module IV Polymorphism	
<p>Descriptors/Topics</p> <ul style="list-style-type: none"> • Polymorphism, Type of Polymorphism – Compile time and runtime, • Function Overloading, • Operator Overloading (Unary and Binary) • Polymorphism by parameter, • Pointer to objects, • this pointer, • Virtual Functions, • pure virtual functions. 	20
Module V Strings, Files and Exception Handling	
<p>Descriptors/Topics</p> <ul style="list-style-type: none"> • Manipulating strings, • Streams and files handling, • formatted and Unformatted Input output. • Exception handling, • Generic Programming – function template, class Template • Standard Template Library: Standard Template Library, Overview of Standard Template Library, • Containers, Algorithms, Iterators, <p>Other STL Elements, The Container Classes, General Theory of Operation, Vectors</p>	20

Student Learning Outcomes:

1. To apply the knowledge gained in areas of Information Technology, Computing, Mathematics, and Basic Science & Engineering. Analyze and attract the vital resources required to turn a vision into reality.
2. To build a robust foundation in theoretical and experimental work to analyse, create and design software products, processes and systems. Experimental work to analyse, create and design software products, processes and systems.
3. To gain expertise in designing, implementation and development of computer based systems and IT processes.
4. To use current techniques, skills, and tools necessary for computing practice.

Pedagogy for Course Delivery:

The class will be taught using theory and case based method. In addition to assigning the case studies, the course instructor will spend considerable time in understanding the concept of innovation through the eyes of the consumer. The instructor will cover the ways to think innovatively liberally using thinking techniques.

Lab/ Practicals details, if applicable:

List of Experiments:

1. Simple C++ Programs to Implement Various Control Structures.
 - a. If statement
 - b. Switch case statement and do while loop

c. For loop

d. While loop

2. Programs to Understand Structure & Unions.

a. Structure

b. union

3. Programs to Understand Pointer Arithmetic.

4. Functions &

Recursion. a. Recursion

b. function

5. Inline Functions.

6. Programs to Understand Different Function Call Mechanism. a. Call by reference & Call by Value

7. Programs to Understand Storage Specifiers.

8. Constructors & Destructors.

9. Use of “this” Pointer. Using class

10. Programs to Implement Inheritance and Function Overriding. a. Multiple inheritance –Access Specifiers

b. Hierarchical inheritance – Function Overriding /Virtual Function

11. Programs to Overload Unary & Binary Operators as Member Function & Non Member Function.

a. Unary operator as member function

b. Binary operator as non member function

12. Programs to Understand Friend Function & Friend

Class. a. Friend Function

b. Friend class

13. Programs on Class Templates

Assessment/ Examination Scheme:

Theory L/T (%)	Lab/Practical/Studio (%)	End Term Examination
75	25	100

Theory Assessment (L&T):

Continuous Assessment/Internal Assessment					End Term Examination
Components (Drop down)	CT	HA/P	S/V/Q	A	EE
Weightage (%)	10	7	8	5	70

Lab/ Practical/ Studio Assessment:

	Continuous Assessment/Internal Assessment				End Term Examination		
Components (Drop down)	PRACTICAL / LAB RECORD	PERFORMANCE	VIVA VOCE	Attendance	PR	V	Total
Weightage (%)	10	10	5	5	35	35	70

Text:

- R. Lafore, “Object Oriented Programming using C++”, BPB Publications, 2004
- “Object Oriented Programming with C++” By E. Balagurusamy

References:

- A.R. Venugopal, Rajkumar, T. Ravishanker “Mastering C++”, TMH, 1997
- Schildt Herbert, “C++: The Complete Reference”, Wiley DreamTech, 2005