

Paper Code(s): CIC-211	L	P	C
Paper: Object-Oriented Programming Using C++	4	-	4

Marking Scheme:												
1. Teachers Continuous Evaluation: 25 marks												
2. Term end Theory Examinations: 75 marks												
Instructions for paper setter:												
1. There should be 9 questions in the term end examinations question paper.												
2. The first (1 st) question should be compulsory and cover the entire syllabus. This question should be objective, single line answers or short answer type question of total 15 marks.												
3. Apart from question 1 which is compulsory, rest of the paper shall consist of 4 units as per the syllabus. Every unit shall have two questions covering the corresponding unit of the syllabus. However, the student shall be asked to attempt only one of the two questions in the unit. Individual questions may contain upto 5 sub-parts / sub-questions. Each Unit shall have a marks weightage of 15.												
4. The questions are to be framed keeping in view the learning outcomes of the course / paper. The standard / level of the questions to be asked should be at the level of the prescribed textbook.												
5. The requirement of (scientific) calculators / log-tables / data – tables may be specified if required.												
Course Objectives :												
1.	To introduce the basic Concepts of Object Oriented Programming (data types, operators and functions) using C++											
2.	To introduce concepts of Classes and Objects with the examples of C++ programming											
3.	To understand object oriented features such as Inheritance and Polymorphism											
4.	To use various object oriented concepts (exceptional handling) to solve different problems											
Course Outcomes (CO)												
CO 1	Ability to have an in-depth knowledge of object oriented programming paradigm											
CO 2	To be able to develop basic C++ programming skills											
CO 3	To be able to apply various object oriented features using C++											
CO 4	Ability to have an understanding of generic programming & standard templates											
Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)												
	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12
CO 1	3	2	2	2	3	-	-	-	3	2	2	3
CO 2	3	2	2	2	3	-	-	-	3	2	2	3
CO 3	3	2	2	2	3	-	-	-	3	2	2	3
CO 4	3	2	2	2	3	-	-	-	3	2	2	3
UNIT – I												
Object Oriented Programming Paradigm, Basic Concepts of Object Oriented Programming, Benefits of Object Oriented Programming, Object Oriented Languages, Applications of Object Oriented Programming, C++ Programming Language, Tokens, Keywords, Identifiers and Constants, Data Types, Type Compatibility, Variables, Operators in C++, Implicit Type Conversions, Operator Precedence, The Main Function, Function Prototyping, Call by Reference, Return by Reference, Inline Functions, Function Overloading, Friend Functions, default parameter value.												
UNIT – II												
Specifying a class, Member Functions, Encapsulation, information hiding, abstract data types, objects & classes, Static Member Functions, Arrays of Objects, Constructors & Destructors, Parameterized Constructors, Copy Constructors, Dynamic Constructors, Destructors, identity and behaviour of an object, C++ garbage collection, dynamic memory allocation, Explicit Type Conversions, Operator Overloading.												
UNIT – III												
Inheritance, inheritance methods, Class hierarchy, derivation – public, private & protected, aggregation,												

Inheritance Constructors, composition vs. classification hierarchies, Containership, Initialization List, Polymorphism, categorization of polymorphic techniques, polymorphism by parameter, parametric polymorphism, generic function – template function, function overriding, run time polymorphism, virtual functions.

UNIT – IV

Standard C++ classes, using multiple inheritance, persistent objects, streams and files, namespaces, exception handling, generic classes, standard template library: Library organization and containers, standard containers, algorithm and Function objects, iterators and allocators, strings, streams, manipulators, user defined manipulators, vectors.

Textbook(s):

1. Stanley B. Lippman, Josée Lajoie, Barbara E. Moo, "C++ Primer", Addison-Wesley Professional, 2012.
2. Ivor Horton, "Using the C++ Standard Template Libraries", Apress, 2015.
3. R. Lafore, "Object Oriented Programming using C++", Galgotia.

References:

1. A.R.Venugopal, Rajkumar, T. Ravishanker "Mastering C++", TMH
2. Bjarne Stroustrup, "Programming: principles and practice using C++", Addison-Wesley, 2015.
3. Bjarne Stroustrup, "A Tour of C++", Addison-Wesley Professional, 2018.
4. Bjarne Stroustrup, "The C++ Programming Language", 4th Edition, Addison-Wesley Professional, 2013.
5. Peter Van Weert and Marc Gregoire, "C++17 Standard Library Quick Reference: A Pocket Guide to Data Structures, Algorithms, and Functions", Apress (2019)
6. Rumbaugh et. al. " Object Oriented Modelling & Design", Prentice Hall
7. G . Booch "Object Oriented Design & Applications", Benjamin,Cummings.
8. E.Balaguruswamy, "Objected Oriented Programming with C++", TMH
9. Steven C. Lawlor, "The Art of Programming Computer Science with C++", Vikas Publication.
10. Slobodan Dmitrović, "Modern C++ for Absolute Beginners":A Friendly Introduction to C++ Programming Language and C++11 to C++20 Standards", Apress, 2020.