CS - 14 : C++ and Object Oriented Programming

Objectives:

- To provide of OOPs concepts, input/output data management, arrays in C++, functions, classes, objects, pointers, and much more.
- Object-Oriented features, which allow the programmer to create objects within the code.

Prerequisites:

concepts of OOPs and their implementation.

No	Topics	Details	Marks weight in %	Min. Lect.
1	**Principles of object oriented programming Tokens, expressions and control statements**	• Procedure – oriented programming • Object oriented programming paradigm • Basic concepts of object oriented Programming • Benefits of object oriented programming • Application of object oriented programming • What is c++? • Application of c++ • Input/output operators • Structure of c++ program • Introduction of namespace • Tokens : keywords, identifiers, basic data types, user- defined types, derived data types, symbolic constants, type compatibility, declaration of variables, dynamic initialization of variables, reference variables • Operators in C++: scope resolution operator, member referencing operator, memory management operator, manipulators, type cast operator. • Expression : Expression and their types, special assignment operator, implicit conversions, operator precedence	20	15

		• Control structures ♦ Conditional control structure :- simple if, if…else , nested if else, switch etc. ♦ Looping control structure:- for, while , do…while		
	Functions in C++	• The main function • Function prototype • Call by reference • Return by reference • Inline function • Default arguments • Const arguments • Functions overloading • Adding C Functions turbo C++		
2	**Classes and Objects, Constructor and Destructor**	• C structures revisited • Specifying a class • Local Classes • Nested Classes • Defining member functions, nesting of Member functions, private member function, making outside function inline • Arrays within a class • Memory allocation for objects • Static data member • Static member functions • Arrays of objects • Objects as function arguments • Friendly functions • Returning objects • Const member function • Pointer to members	20	12
		• Characteristics of constructor • Explicit constructor • Parameterized constructor • Multiple constructor in a class • Constructor with default argument • Copy constructor • Dynamic initialization of objects • Constructing two dimensional array		

		• Dynamic constructor • MIL , Advantage of MIL • Destructors		
3	**Operator overloading and type conversion, Inheritance**	• Concept of operator overloading • Over loading unary and binary operators • Overloading of operators using friend Function • Manipulation of string using operators • Rules for operator overloading • Type conversions. • Comparison of different method of conversion • Defining derived classes • Types of inheritance (Single, Multiple, Multi-level, Hierarchical, Hybrid) • Virtual base class & Abstract class • Constructors in derived class • Application of Constructor and Destructor in inheritance • Containership, Inheritance V/s Containership	20	11
4	**Pointer, Virtual functions and Polymorphism, RTTI Console I/O operations**	• Pointer to Object • Pointer to derived class • this pointer • Rules for virtual function • Virtual function and pure virtual function. • Default argument to virtual function • Run Time Type Identification • C++ streams • C++ stream classes • Unformatted and formatted I/O operations • Use of manipulators.	20	10
5	**Working with Files, Exception handling,**	• File stream classes • Opening and closing a file • Error handling • File modes	20	12

	Introduction to Template STL	• File pointers • Sequential I/O operations • Updating a file (Random access) • Command line arguments • Overview of Exception Handling • Need for Exception Handling • various components of exception handling • Introduction to templates • Class templates • Function templates • Member function templates • Overloading of template function • Non-type Template argument • Primary and Partial Specialization • Introduction to STL • Overview of iterators, containers		
TOTAL			100	60

Students seminar - 5 Lectures.
Expert Talk - 5 Lectures
Students Test - 5 Lectures.
TOTAL LECTURES 60+15=75

Course outcomes:

- Understand the concept and underlying principles of Object-Oriented Programming.
- Understand implementation issues related to object-oriented techniques.
- Apply the techniques of object-oriented programming to solve real problems
- Analyze, apply and write programs that make appropriate use of object-oriented functionality such as classes, overloading and inheritance
- Implement the file handling techniques for back-end storage problems solutions

Reference Books:

1. Complete Reference C++ by Herbert Schildt McGraw Hill Publications
2. Computer Science- A Structured approach using C++ by Forouzan, Gilburg, THOMSON
3. Object Oriented Programming in C++ - E.Balagurusamy, BPB
4. Object Oriented programming in C++ by Robert Lafore, Pearson Education
5. Mastering C++ - Venugopal
6. The C++ Programming Language by Bjarne Stroustrup, Pearson Education
7. Object Oriented Programmin in C++ - Robaret Laphore
8. Let us C++ - Yashvant Kanitkar, BPB