Day wise course coverage
Object Oriented Programming (IT-2005)

Day	Topics to be covered
1	**Evolution of programming paradigm:** - Procedure oriented programming Vs. Object-oriented programming (OOP) - Q&A
2	**Object oriented programming concepts:** - Objects - Classes - Encapsulation and abstraction, - Inheritance - Polymorphism, - Dynamic binding - Message passing - Q&A
3 4	**Review of fundamental constructs of C used in C++:** - Character set - Keyword - Constant - Variable - Data types - Operator & expression - Control structure (branching & looping) - Typecasting - Array & strings - Q&A
5 6	**C++ Programming basics:** - Character set - Keyword - Constant - Data types - Operator & expression - Control structure (branching & looping) - Typecasting - Array & strings - Streams based I/O - Type conversions and casting - Name space - Scope resolution operator (::) - Q&A
7	**Variables:** - Scope & lifetime of variables - Variable declaration at the point of use - Ordinary Variable Vs. Pointer Variable Vs. Reference Variable (variable aliases) - Q&A
8	**Function:** - Function: Parameter passing (i) by value, (ii) by address, (iii) by reference - Inline function - Function overloading - Default arguments - Q&A
9	**Classes and Object:** - Defining class with functions and data members

	- Q&A
10	**Classes and Object:** - Creating & deleting objects by using new and delete operators respectively - Array of Objects - Objects as function argument. - Q&A
11	**Classes and Object:** - Static Data members and member functions. - Q&A
12	**Classes and Object:** - Function with default arguments - Function overloading - Concept of friend function and friend class - Q&A
13	**Constructor:** Definition of constructors and its uses
14	**Types of constructors:** - Default constructor - Parameterized constructor - Copy constructor - Constructor with dynamic allocation - Dynamic constructors
15	Constructor Overloading and Destructors
16	**Inheritance:** - Concept of inheritance: defining derived and base classes - Class hierarchies, - Public, private, and protected derivations
17	**Inheritance:** Types of Inheritance: Single Inheritance, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance, Hybrid Inheritance
18	**Inheritance:** Virtual base class: Function overriding
19	**Inheritance:** Constructors/Destructors in derived classes
20	**Inheritance:** Constructors invocation and data members initialization in derived classes
21	**Inheritance:** Member classes: classes within classes
22	**Polymorphism: Operator overloading:** Introduction
23	**Operator overloading:** - Overloading unary operators, binary operators, - Overloading binary operators using friend function and member function.
24	**Operator overloading:** Rules for overloading operators
25	**Polymorphism**: - Introduction to pointers: Pointers to objects, pointer to derived class object, - this pointer
26	**Polymorphism:** Compile time polymorphism: Review of Function Overloading and Operator overloading
27	**Polymorphism**: Run time polymorphism: virtual functions, pure virtual functions, abstract class, virtual constructors and destructors.
28	**Files and Streams:** - Introduction to file handling: text file Vs. binary file - Hierarchy of file stream classes: Functions of File Stream classes
29	**Files and Streams: Steps to process a File in a program. Different functions used in file** - Steps to process a File in a program:

	- Create an stream object (input or output or i/o) by declaring the stream to be of appropriate class - Associate a file with this stream object that is to open the file by using constructor or by using Open() function. - Process the file - Closing the file by using close() function
30	File modes(Sequential and random) **Sequential access:** - The get(), getline() and put() functions - The read() and write() functions - Reading and writing class objects
31	**Files and Streams:** - **File pointers and their Manipulations:** two file pointers (get pointer, put_pointer), Functions for manipulation of file pointers (seekg(), seekp(), tellg(), tellp()) - **Updating a File:** Random Access: - **Error handling during file operation:** Error handling functions (eof(), fail(), bad(), good())
32	**Exception Handling:** Basics of Exception Handling
33	**Exception Handling:** Exception Handling Mechanism: The keyword try, throw and catch
34	**Templates:** Need of template
35	**Templates:** Class Templates: Definition, Class Template with multiple parameters
36	**Templates:** Function Templates: Definition, Function Template with multiple parameters