Ques. No.	Unit I
1.	Distinguish between procedure-oriented programming and object-oriented programming.
2.	Explain various features of OOP with illustrations.
3.	Discuss advantages/benefits of object oriented programming.
4.	Discuss disadvantages of procedural/conventional programming.
5.	Explain the key concepts of OOP.
6.	Distinguish between C and C++
7.	Discuss the applications of OOP.
8.	Explain in detail what is polymorphism with suitable example.
9.	
10.	**Unit II**
11.	Explain how a class is defined and instantiated with example.
12.	Write a program to create Student class to store student_id, student_name and registration_no. Display the information for 3 students by using appropriate method.
13.	Write a program to create Person class to store person_name, person_age and address. Display the information for 3 persons by using appropriate method.
14.	Write a class for Student having properties student_rno and student_name. Initialize one object of Student by using constructor with the default value. Initialize the second object by passing values to the parameterized constructor. Print both the objects using appropriate method.
15.	Explain the concept of function overloading and conditions of overloading with suitable example.
16.	Apply the concept of function overloading for calculating Volume of cylinder and cube.
17.	Apply the concept of function overloading for calculating area of any two shapes.
18.	In a super mart, an owner wanted to calculate the area of the each compartments which are in rectangular, cylindrical and square shape. Develop a C++ program to implement this scenario.
19.	Apply the concept of constructor to the following scenario: Write a class for Car having properties model_year and model_name. Initialize one object of Car by using constructor with the default value. Initialize the second object by passing values to

	the parameterized constructor. Print both the objects using appropriate method.
20.	Distinguish between Constructor and Destructor.
21.	Explain in detail access specifiers in C++ with suitable examples.
22.	Explain characteristics of constructors and its types with suitable example.
23.	Explain in detail destructors with suitable example.
24.	Explain the properties of anonymous object with suitable example.
25.	Apply the concept of passing the object to calculate the average marks of two students.
26.	Compare and contrast Constructor and Destructor.
27.	Write a C++ program to calculate average marks of two students. Display the average using suitable method.
28.	Apply the concept of function overloading to find area of circle, square and rectangle. Display the area using suitable method.
29.	**Unit III**
30.	Write a program in C++ to find the smaller number from two numbers using friendly function. (Use two classes and one friend function which can access the private members of both the classes.)
31.	Explain in detail friend functions with suitable example
32.	Write a friend function named 'Result' to add private data members SUB1, SUB2, SUB3 of class Marks and EXTRA of class Cocurricular. Student should be declared as pass if he scores more then 40 marks in every subject. Percentage should be calculated using marks of both the classes. If student is fail, display the count of subjects in which student failed.
33.	Apply the concept of returning an object through function: Write a program to add two complex numbers A & B to produce third complex number C and display all the three numbers.
34.	Apply the concept of operator overloading: Write a program to add two complex numbers A & B to produce third complex number C and display all the three numbers.
35.	Write a program in C++ to obtain the sum of two distances given in feet and inch. Overload the operator ‘+’ for obtaining the sum, use constructor for storing the data.
36.	Apply the concept of operator overloading to overload the unary – (minus) operator to negate the value of three variables also display them.
37.	Apply the concept of inheritance to the following scenario: Create a class Calculation having addition () and subtraction()

	methods. Inherit this class in another class MyCalculation having multiplication() method. Access all the methods using derived class object.
38.	Assume that test results of a batch of students are stored in three different classes. Class student stores the roll-number, class test stores the marks obtained in two subjects and class result contains the total marks obtained in the test. The class result can inherit the details of marks obtained in the test and the roll-number of students through multilevel inheritance. Write a program in C++ for above assumption.
39.	Discuss advantages and disadvantages of inheritance.
40.	Explain in detail friend functions with suitable example
41.	**Unit IV**
42.	Explain what are pure virtual functions.
43.	Apply the concept of run time polymorphism to following scenario: Consider a book shop which sells both books and video tapes. Create a class media that stores the title and the price of publication. Create two derived classes one for storing the number of pages in the book and another for storing the playing time of the tape. A function display() is used in all the classes to display the class contents.
44.	Distinguish between Compile time polymorphism and run time polymorphism.
45.	Explain what are virtual functions with suitable example.
46.	Explain in brief what is polymorphism and types of polymorphism.
47.	**Unit V**
48.	Write a generic program to add two numbers of two different data types using function templates.
49.	Explain how C++ handles any exception in detail.
50.	Apply the concept of function template to swap two numbers of two different data types.
51.	Write a program for handling the exception Invalid Age that generates exception whenever age is invalid. (Consider valid age as 18 years and above).
52.	Apply the concept of function template to add two numbers of integer and double data types.
53.	Write a program with the following: 1. A function to read two integer numbers from keyboard. 2. A function to calculate the division of these two numbers. 3. A try block to detect and throw an exception if the condition

	“divide-by-zero” occurs. 4. Appropriate catch block to handle the exceptions thrown.
54.	**Unit VI**
55.	Explain what is standard template library in detail.
56.	Discuss the components of standard template library.
57.	Explain different containers of standard template library.
58.	Distinguish between lists and vectors.
59.	Explain different containers of standard template library.