Unit-2 Fundamental Concepts of OOP with C++

1. Describe any two characteristics of OOP.
2. Explain the benefits of Object Oriented Programming.
3. Explain the difference between structure and class in C++.
4. Compare structured programming with object oriented programming.

Unit-3 C++ Programming Syntactical Basics

1. Explain >> and << operators.
2. Explain manipulators.
3. Discuss new and delete keywords.
4. Describe, with examples, the uses of enumeration data types.
5. List the types of comments available in C++.

Unit-4 C++ Functions

1. Explain merits and demerits of inline functions.
2. Explain default arguments.
3. Explain friend function with the help of an example.
4. What do you mean by call by value and call by reference? Justify your answer with suitable example.
5. Describe The Following Terms In C++ With Example
   i) Default Argument. ii) Function Overloading. iii) Destructor iv) this pointer
6. What is friend function? Write a program to find out sum of two private data members x and y of two classes A and B using a common friend function. Assume that the prototype for both the classes will be void sum (A, B)
7. Explain function overloading with a program.

Unit-5 Objects and Class

1. Explain how to allocate and de-allocate memory dynamically.
2. Explain keywords private, protected and public.(access specifiers in C++)
3. Explain needs and usage of this pointer. Explain with suitable program.
4. Explain copy constructor and its use by giving an example.
5. What are the characteristics of constructor in C++?Explain the default Constructor.
6. Write down the uses of static data member and static member function with suitable example. When do we declare a member of a class static?
7. Discuss nameless temporary object.
8. Explain Scope Resolution Operator (::) with a program.

9. What is a User Defined Data type? What is the scope and life time of variable? Explain using C++ program.
10. Describe the importance of destructors.
11. Discuss the different ways by which we can access public member functions of an Object.
12. Differentiate constructor overloading and function overloading.

Unit-6 Operator Overloading

1. Is assignment operator overloaded by default? Justify your answer. When do you overload it mandatorily?
2. What is an operator function? Write a program to overload binary + operator as a member function.
3. Create a class coordinate containing x, y and z private variables. Perform operations for incrementing, adding and comparing object(s) by overloading + +, + = and = = operators respectively. Define necessary functions to set and display the variables.
4. How many arguments are needed to overload a unary operator and a binary operator. (Consider the operator overloading function as a member function of the class).
5. Explain type conversion from class type to basic type and one class type to another class type with suitable example.

Unit-7 Inheritance

1. Describe multi-level and multiple inheritances.
2. What is inheritance? Why do we use of protected specifier for base class data members in inheritance? Explain hybrid inheritance with example.
3. Describe the syntax of the single inheritance in C++.
4. Differentiate between multiple inheritance and multi-level inheritance.

Unit-8 Polymorphism & Virtual Functions

1. Explain two usage of keyword virtual.
2. What is polymorphism? How is it achieved in C++? Explain with suitable example.
3. What is an abstract class? Explain virtual base class with a program.
4. Demonstrate the use of virtual functions with a program.
5. Explain Runtime polymorphism (Function overriding). Explain and demonstrate, how virtual function to achieve runtime polymorphism?
6. Explain the difference between virtual function and virtual base class.

Unit-9 Templates and Exception

1. By giving an example, illustrate use and working of nested try blocks and re -throwing of an exception.
2. Explain try, catch and throw. Give one simple example.(Explain with an example like "division by zero" situation.)
3. What is the purpose of using template in C++? Explain template function and template class with example.
4. Explain catch all exception and multiple catch exception with example.
5. Create a generic class stack using template and implement common Push and Pop operations for different data types.
6. Discuss Standard Template Library.
7. How to create generic methods in C++? What are its advantages?

Unit-10 Introduction to Streams and Files

1. Explain tellg and seekp functions.
2. What is a stream? Describe various stream classes for console I/O operations in C++.
3. Explain the use of binary files in C++ with a program.
4. Explain ios::app and ios::ate flags.
5. Explain ios::binary and ios::trunc flags.
6. How do the I/O facilities in C++ differ form that in C?
7. Write a program which scans the five integer values from a file and print them on the output window.

Short Questions

1. Why C++ is not a pure object oriented programming language?
2. Explain encapsulation property of an OOP language.
3. Define *class* in C++.
4. What is the use of namespace in C++?
5. What are *stream extraction* and *stream insertion* operators?
6. Define *destructor*.
7. What is a *friend class*?
8. What is an *abstract* class?
9. What is the use of private inheritance?
10. Briefly explain multi-level inheritance.
11. Explain *pure virtual function*.
12. Discuss various *access specifiers* in C++.
13. What is the use of overloading an assignment (=) operator?
14. Explain *tellg( )* function in file streams.
15. What is an object?
16. Define Data Abstraction.
17. What is a constructor?
18. What is the purpose of 'delete' operator?
19. Why do we need the preprocessor directive #include<iostream>?
20. Define Reusability.
21. What does this pointer point do?
22. Define std::cout and use it in a C++ statement.
23. What is the difference between an array and a simple variable?
24. How does a C++ Structure differ from a C++ Class?
25. Can We have more than one constructor in a Class?
26. Enlist Derived data types in C++.
27. How is a member function of a class defined?
28. What is a reference variable?
29. When will you make a function inline?
30. When do we declare a member of a class static?
31. How do we invoke a constructor function?
32. What is an operator function?
33. When do we use protected visibility to a class member?
34. When do we make a class virtual?
35. What does this pointer do?
36. What does the iomanip file play?
37. Write a statement using seekg() to go backward by 20 bytes end.
38. When should a program throw an exception?

1. Create a class *coordinate* containing x, y and z private variables. Perform operations for incrementing, adding and comparing object(s) by overloading + +, + = and = = operators respectively. Define necessary functions to set and display the variables.
2. Write a program to create a class distance containing feet and inches. Using operator keyword, convert an object of class distance into total meters which is a float data type. (1 meter=3.28 feet)
3. Declare a class called book having members like book_title, publisher and author_name. Overload extractor and inserter operators (>> and <<) for class book.
4. Write a program with class Teacher which has the data members: name and age.From the class Teacher inherit a class Professor which has the data member: degree and a function display. The display function prints the values of name, age and degree for an object. Create the object of class Professor in main() method and call the display method for it. Initialize the values thru the use of appropriate constructor.

5. Write a program for runtime polymorphism with the following definition : Create a class Employee with the members name and basic. Define a pure virtual function compute_salary. The employee class should be extended by two other classes namely:
   Manager and Worker. In manager, the compute_salary will compute the salary as cube of basic and in worker the compute_salary will compute the salary as double of basic. In main method create a pointer of the Employee class, which first points to Manager object and compute his salary, then the same object points to Worker object and compute his salary.