Class and Object (1–15)

1. Write a C++ program to create a class Student with data members name and rollNo. Create and display objects.
2. Create a class Rectangle with length and breadth. Write functions to calculate area and perimeter.
3. Create a class Car with data members brand, model, and year. Display data using a member function.
4. Define a class BankAccount with account number and balance. Include a method to deposit money.
5. Create a class Complex for complex numbers. Initialize and display complex numbers.
6. Define a class Book with title, author, and price. Use a function to set and display data.
7. Create a class Time with hours, minutes, seconds. Write a function to display time in HH:MM:SS format.
8. Define a class Circle to find area and circumference.
9. Create a class Employee with data members ID, name, salary. Use functions to accept and show details.
10. Create a class Date and write a method to validate and display it.
11. Define a class Calculator with methods to add, subtract, multiply, and divide.
12. Write a program to implement class with private, public, and protected access specifiers.
13. Create a class Movie with attributes title, director, and year. Display movie info.
14. Define a class Point with x and y coordinates. Display the point.
15. Create a class Person and read and display information of multiple persons using array of objects.

Constructor (16–25)

16. Write a program to demonstrate default constructor.
17. Create a class Box and define a parameterized constructor to initialize its dimensions.
18. Implement a copy constructor in a class Item.
19. Define a class Fraction with default and parameterized constructors.
20. Create a class Laptop with a constructor that prints "Object Created".

21. Implement constructor overloading in class Room.
22. Use initializer list in a constructor.
23. Create a constructor that takes another object as argument.
24. Demonstrate constructor with default arguments.
25. Create a class Student where constructor reads data from the user.

Method Overloading (26–32)

26. Write a program to overload a function area() for circle, rectangle, and square.
27. Overload a method print() to print int, float, and string.
28. Create a class Math with overloaded sum() method for different parameter types.
29. Overload volume() method to calculate volume of cube and cuboid.
30. Demonstrate function overloading for a method that calculates interest (simple & compound).
31. Implement method overloading in a class Temperature to convert Celsius to Fahrenheit and vice versa.
32. Overload a method max() to return maximum of two or three numbers.

Static Data Members and Member Functions (33–38)

33. Create a class Account with a static member to count number of accounts created.
34. Write a program to demonstrate use of static member function.
35. Create a class where a static variable is used to assign unique IDs to objects.
36. Use static member to share a value across all objects.
37. Demonstrate access of static data without creating an object.
38. Write a program with static counter to count number of objects destroyed.

Operator Overloading (39–51)

39. Overload + operator for a class Complex.
40. Overload - operator for class Vector.
41. Overload * operator for a class Matrix.

42. **Overload == operator for comparing two strings.**

43. **Overload ++ operator (both prefix and postfix) for a class Counter.**

44. **Overload << and >> operators for a class Student.**

45. **Overload relational operators <, > for class Distance.**

46. **Create a class Currency and overload + and -.**

47. **Overload the assignment operator =.**

48. **Overload the indexing operator [] for a class Array.**

49. **Overload the function call operator ().**

50. **Overload the != operator for a class.**

51. **Overload the >= and <= operators in a class Score.**

Friend Function (52–58)

52. **Create a friend function to access private members of class Box.**

53. **Use a friend function to calculate the sum of two objects from class Complex.**

54. **Demonstrate friend function between two different classes.**

55. **Write a program where friend function modifies private members of class.**

56. **Implement a friend function to compare two private members of different classes.**

57. **Use a friend function to swap data of two classes.**

58. **Create a friend class and show how it accesses another class's private members.**

Inheritance (59–72)

59. **Demonstrate single inheritance with class Animal and class Dog.**

60. **Implement multilevel inheritance with Person -> Employee -> Manager.**

61. **Use hierarchical inheritance with a base class Shape and derived classes Circle, Rectangle.**

62. **Show multiple inheritance with Father and Mother classes to create Child.**

63. **Demonstrate hybrid inheritance with proper structure.**

64. **Write a program to access base class constructor from derived class.**

65. **Use protected members in base class and access from derived class.**

66. Override a method from base class in derived class.

67. Call base class function from derived class.

68. Show ambiguity in multiple inheritance and resolve it using scope resolution.

69. Use constructors in base and derived class and show the order of execution.

70. Demonstrate use of virtual base class.

71. Inherit private data members and access them using public methods.

72. Create a class Vehicle and derive Car and Bike. Show polymorphism using inheritance.

Method Overriding (73–77)

73. Write a base class with a virtual function and override it in a derived class.

74. Demonstrate method overriding in derived class with different implementation.

75. Use override keyword to explicitly override a base class function.

76. Call base class overridden method using scope resolution.

77. Implement a real-world scenario of method overriding in a Shape class hierarchy.

Virtual Function (78–85)

78. Create a virtual function in base class and override in derived.

79. Demonstrate run-time polymorphism using virtual functions.

80. Write a program to show virtual function call through base class pointer.

81. Use a virtual destructor and show its importance.

82. Create an abstract class with pure virtual function.

83. Implement multiple derived classes with overridden virtual functions.

84. Demonstrate use of virtual functions in class hierarchy Media -> Book, Tape.

85. Show use of virtual functions for interface design.

File Handling (86–100)

86. Write a program to create and write data to a file.

87. Read data from a text file and display it.

88. Append data to an existing file.

89. Write and read a class object to/from file using ofstream and ifstream.

90. Count number of words in a file.

91. Copy content from one file to another.

92. Write multiple student records to a file and then read all.

93. Use seekg() and tellg() to navigate a file.

94. Delete a specific record from a file.

95. Update data of a specific record in a file.

96. Display records of all students with marks above a threshold from file.

97. Write a program to store login credentials in file and verify login.

98. Count lines, words, and characters in a text file.

99. Store employee data in a binary file and retrieve based on employee ID.

100. Create a program to write and read inventory items from file using fstream.