Practical File

1. Create a class Student to store roll number, name, and marks; include methods to input and display.
2. Write a class Book with parameterized and default constructors.
3. Implement a class BankAccount with constructors and functions to deposit and withdraw.
4. Demonstrate the use of constructor overloading with a Time class.
5. Create a class Rectangle with private members, use this pointer to resolve naming conflict.
6. Design a class with a destructor that prints when an object is destroyed.
7. Use copy constructor to create a deep copy of an object.
8. Create a class Date and initialize using an initializer list in constructor.
9. Write a class to convert temperature between Celsius and Fahrenheit using constructor.
10. Create a class with dynamic memory allocation using constructor and destructor.
11. Write overloaded functions to find the maximum of two, three, and four numbers.
12. Create a class Complex and overload the + operator to add two complex numbers.
13. Overload the == operator to compare two objects of a class Person based on age.
14. Overload << and >> operators to input/output a class Distance.
15. Overload the unary - operator to negate a Vector class.
16. Overload the [] operator in a class Array to access elements.
17. Overload ++ operator (prefix and postfix) for a class Counter.
18. Implement function overloading to calculate area of circle, rectangle, and triangle.
19. Write a program to overload * operator to multiply two matrices using class.
20. Overload assignment operator (=) for deep copy in a class with pointer data member.
21. Implement single inheritance using a base class Person and derived class Employee.
22. Create a base class Shape and derive Rectangle and Circle to demonstrate hierarchical inheritance.
23. Show multilevel inheritance using Person -> Employee -> Manager.
24. Demonstrate multiple inheritance using a class Result inheriting from Test and Sports.
25. Create a class hierarchy where Vehicle is base, Car and Bike are derived.
26. Write a program to show constructor call order in multilevel inheritance.
27. Show use of protected members in base and access from derived class.
28. Use virtual base class to resolve ambiguity in hybrid inheritance.

29. Create a program to invoke base class function from derived class explicitly.
30. Demonstrate constructor chaining in inheritance.
31. Create a class Shape with virtual function area() and override in derived classes.
32. Demonstrate function overriding with a base and derived class.
33. Write a program to achieve runtime polymorphism using virtual functions.
34. Create a class with pure virtual function and implement abstract class.
35. Show use of virtual destructors in base and derived classes.
36. Write a program where base class pointer points to derived class object and calls virtual function.
37. Implement runtime polymorphism with multiple derived classes calling a base function.
38. Use virtual function table (vtable) concept to simulate polymorphism.
39. Create a program using abstract class Employee and derived classes Engineer, Manager.
40. Create a virtual function that is overridden to return different shapes' area values.
41. Implement a class with static data member and static function.
42. Create a class to demonstrate friend function accessing two different classes.
43. Use const member functions in a class to restrict modification of data members.
44. Write a program to demonstrate the difference between shallow copy and deep copy.
45. Implement inline member functions inside class definition.
46. Create a class and make a friend class access its private members.
47. Demonstrate encapsulation using private and public access specifiers.
48. Implement a class with overloaded constructor and destructor printing a message.
49. Create a singleton class to allow only one object of the class.
50. Demonstrate object slicing in inheritance using a base and derived class.