PART 1: Class, Objects, Methods, Variables (1-10)

1. Create a class Laptop with attributes: brand, processor, ram. Create objects and print their details.
2. Write a class Employee with a method display_info() to print employee name and salary.
3. Demonstrate instance variables and instance methods using a Student class.
4. Demonstrate class variable (shared by all objects) using a class Company with a class variable company_name.
5. Demonstrate local variables inside a method using a class Demo.
6. Create a class Calculator with methods for addition, subtraction, multiplication, and division.
7. Write a class method using @classmethod to display class-level data.
8. Write a static method using @staticmethod to print a general welcome message.
9. Create a constructor (__init__) to initialize object attributes and display them.
10. Demonstrate the __str__() method to show a meaningful string output when printing an object. ()

PART 2: Inheritance (11-18)

11. Demonstrate Single Inheritance with a class Vehicle and derived class Car.
12. Demonstrate Multilevel Inheritance with classes Person → Employee → Manager.
13. Demonstrate Multiple Inheritance with classes Mother, Father, Child.
14. Demonstrate Hierarchical Inheritance with a base class Shape and child classes Circle, Rectangle.
15. Write a program showing Hybrid Inheritance with a combination of above inheritance types.
16. Demonstrate method overriding using a base class Animal and derived class Dog.
17. Create constructors in parent and child classes and show constructor calling sequence.
18. Demonstrate use of super() keyword to call parent class methods from child class.

PART 3: Polymorphism (19-24)

19. Create different classes with the same method name area(), showing polymorphism.
20. Write a program to demonstrate Duck Typing in Python.

21. Demonstrate operator overloading using __add__ method between two objects.

22. Write a class Animal with method make_sound(). Override it in child classes like Dog,Cat,Cow

23. Demonstrate method overloading using default arguments (Python style).

24. Write a program to show polymorphism through functions that accept different class objects.

PART 4: Encapsulation (25-27)

25. Create a class BankAccount with private attributes and access them through methods (deposit, withdraw).

26. Create a class with getter and setter methods to access and update private variables.

27. Write a program to protect sensitive data like password through encapsulation.

PART 5: Abstraction (28-30)

28. Write a program using ABC and abstractmethod (from abc module) to create an abstract class Vehicle.

29. Create an abstract class Shape with an abstract method area(). Implement it in derived classes Circle, Rectangle.

30. Create an abstract base class Appliance and concrete classes WashingMachine, Refrigerator implementing required methods.