1. Veichle Inheritance

Create a Python program that models a hierarchy of vehicles using inheritance. Start with a base class Vehicle, and then create two or more derived classes (e.g., Car, Bicycle, Motorcycle) that inherit from the Vehicle class. Each class should have specific attributes and methods related to the type of vehicle it represents.

1. Define the Vehicle base class with common attributes such as make, model, and year, and methods like **start()**, **stop(),** and **fuel_up().**

2. Create derived classes for different types of vehicles, e.g., **Car, Bicycle**, and **Motorcycle**. Each derived class should inherit from the **Vehicle** base class and add attributes and methods specific to that type of vehicle. For example, the **Car** class might have attributes like **num_doors**, and the **Bicycle** class could have attributes like **num_gears**.

3. Implement specific methods for each derived class. For instance, the **Car** class might have a method to honk the horn, and the **Bicycle** class could have a method to ring the bell.

4. Create instances of each vehicle type and demonstrate their specific methods and attributes. For example, you can create a car, bicycle, and motorcycle, and call methods like **start(), stop(),** and their specific methods like **honk_horn() or ring_bell().**

2 Polymorphism

Create a Python program that explores polymorphism using a hierarchy of shapes. Start with a base class Shape, and then create two or more derived classes (e.g., **Circle**, **Rectangle**, **Triangle**) that inherit from the Shape class. Each shape class should have its own implementation of methods like **area**() and **perimeter**(). These methods will calculate the area and perimeter of the respective shape.

1. Define the Shape base class with methods like **area**() and **perimeter**(). You can initialize any common attributes in the base class.

2. Create derived classes for different shapes, e.g., **Circle**, **Rectangle**, and **Triangle**. Each derived class should inherit from the **Shape** base class and implement its own version of the **area**() and **perimeter**() methods.

3. Implement specific methods for each derived class. For example, the **Circle** class might have a method to calculate its area based on the radius, and the **Rectangle** class could have a method to calculate its area based on length and width.

   Create instances of each shape type and demonstrate the use of polymorphism by calling the area() and perimeter() methods on them.