1. As a software developer for a zoo simulation system, you need to model different animal behaviors. Create a base class Animal with a virtual function speak(). Derive two classes Dog and Cat, overriding the speak() function to simulate their unique sounds. Write a program that uses pointers to call the appropriate speak() function based on the object type.
2. You are building an e-commerce platform. Create a base class Payment with a virtual function processPayment(). Derive two classes CreditCardPayment and PayPalPayment that override processPayment() to handle the respective payment methods. Use function overriding to ensure the correct payment method is invoked based on user selection.
3. Design a program for calculating areas of different shapes. Create a base class Shape with a virtual function calculateArea(). Derive classes Rectangle and Circle, overriding the function to calculate their respective areas. Use a base class pointer to demonstrate polymorphism in action.
4. You are designing a speed monitoring system for various vehicles. Create a base class Vehicle with a virtual function maxSpeed(). Derive classes Car and Bike that override maxSpeed() to return their specific maximum speeds. Write a program to determine the speed of different vehicles using polymorphism.
5. In a company payroll system, create a base class Employee with a virtual function calculateSalary(). Derive two classes FullTimeEmployee and PartTimeEmployee, overriding calculateSalary() to compute salaries based on hours worked. Use a base class pointer to manage employees polymorphically.
6. You are developing a banking system. Create a base class BankAccount with a virtual function calculateInterest(). Derive classes SavingsAccount and CurrentAccount, overriding the function to calculate interest differently. Demonstrate runtime polymorphism by calling the appropriate calculateInterest() method.
7. In a learning management system, model different types of users. Create a base class User with a virtual function accessPortal(). Derive classes Student and Teacher, overriding the function to define access levels. Write a program to demonstrate how polymorphism manages access for various users.
8. You are tasked with creating a food delivery system. Create a base class DeliveryVehicle with a virtual function deliver(). Derive classes Bike and Drone, overriding the deliver()

function to define the delivery method. Use function overriding to ensure the correct delivery method is called dynamically.

9. Design a simple game with different types of characters. Create a base class GameCharacter with a virtual function attack(). Derive classes Warrior and Mage, overriding attack() to implement their unique attack methods. Use a base class pointer to call the appropriate attack() function during gameplay.
10. In a travel booking system, create a base class Transport with a virtual function bookTicket(). Derive classes Flight and Train, overriding the function to implement booking details for each mode of transport. Demonstrate polymorphism by using a base class pointer to handle ticket booking dynamically.