**1. Scenario: Bank Account Management**

**Question:**Design a BankAccount class in Swift. This class should have properties like accountNumber, accountHolderName, and balance. It should also have methods for:

* Depositing money into the account
* Withdrawing money from the account
* Displaying the account balance
* Transferring money between two accounts

 2. **Scenario: Shopping Cart**

**Question:** Implement a ShoppingCart class in Swift. It should:

* Contain an array of Product objects.
* Allow adding and removing products from the cart.
* Calculate the total cost of the items in the cart.

**3. Scenario: Task Manager**

**Question:**Create a Task class to manage tasks in a task management system. Each task should have:

* A taskName and dueDate.
* A method to mark the task as completed.
* A method to check if the task is overdue (i.e., if the dueDate has passed)

Inheritance

**Question 1: Employee and Manager Class**

You are building an employee management system. You have an Employee class that holds basic employee details such as name, id, and salary. You also need to create a Manager class that inherits from Employee, with additional properties like department and teamSize.

* How would you design these classes using inheritance in Swift?
* Implement a function in the Manager class to print the details of the manager, including the department and team size.

**Question 2: Vehicle and Car Class**

Consider a scenario where you are building a vehicle tracking app. You have a base class Vehicle with properties like make, model, and speed. You need to create a Car subclass that adds specific properties like numberOfDoors and a function to accelerate the car.

* How would you implement this inheritance structure?
* Include a method in the Car class to increase the speed of the car and print out the updated speed.

**Question 3: Animal Class and Subclasses**

You are working on a zoo management system. You need a base class Animal that has properties like name and age, and a method speak() that is overridden by subclasses. The subclasses should be Dog and Cat, each overriding the speak() method with specific behaviors (e.g., bark for dog, meow for cat).

* How would you implement the classes and the inheritance structure in Swift?
* Ensure that the speak() method in both subclasses displays the corresponding sound when called.

**Question 4: Shape Class and Subclasses**

You are building a geometry app and need a base class Shape that has a calculateArea() method. You will create subclasses for Rectangle, Circle, and Triangle, each overriding calculateArea() to calculate the area of the respective shape.

* How would you design the Shape class and the subclasses in Swift?
* Implement the area calculation for each shape.

**Question 5: Bank Account and Savings Account Classes**

You are building a banking application. The BankAccount class has basic account details like balance and a method deposit(). The SavingsAccount class inherits from BankAccount and adds a property interestRate and a method to calculate interest.

* How would you implement inheritance in this scenario?
* Create a function in the SavingsAccount class to calculate and display the interest based on the current balance.