Problem-Based Questions on Inheritance

Question 1: Bank Account System

Design a program for a bank system. Create a base class `BankAccount` with attributes `accountNumber`, `balance`, and a method `deposit(double amount)` that adds money to the balance. Create a derived class `SavingsAccount` that adds an interest rate attribute and a method `addInterest()` to calculate and add interest to the balance. In the `main()` method, create a `SavingsAccount` object, deposit some money, and calculate the updated balance after interest is added.

Question 2: Library Management

Create a base class `Book` with attributes `title`, `author`, and `ISBN`. Add a method `displayDetails()` to print book details. Create a derived class `EBook` that adds an attribute `fileSize` (in MB) and overrides the `displayDetails()` method to include the file size. Write a program to demonstrate polymorphism by calling the overridden `displayDetails()` method on objects of both `Book` and `EBook`.

Question 3: Online Store Discounts

Create a base class `Customer` with attributes `customerId` and `name`. Add a method `calculateDiscount(double amount)` that returns 5% of the amount as the discount. Create a derived class `PremiumCustomer` that overrides `calculateDiscount()` to provide a 10% discount. Write a program to calculate and display the discount for both regular and premium customers on a purchase amount of INR10,000.

Question 4: Vehicle Rental System

Design a class hierarchy for a vehicle rental system. Create a base class `Vehicle` with attributes `vehicleNumber` and `rentPerDay`. Add a method `calculateRent(int days)` to calculate the total rent. Create a derived class `Car` that adds an attribute `seatingCapacity` and overrides the

`calculateRent()` method to add INR500 per day if the seating capacity exceeds 5. Write a program to calculate the total rent for a `Car` object rented for 3 days with a seating capacity of 7.

Question 5: College Staff System

Create a base class `Staff` with attributes `name`, `designation`, and `salary`. Add a method `displayDetails()` to print the staff details. Create two derived classes: 1. `TeachingStaff` with an attribute `subject` and a method `conductClass()`. 2. `NonTeachingStaff` with an attribute `department` and a method `manageWork()`. Write a program to create objects of both `TeachingStaff` and `NonTeachingStaff`, display their details, and call their respective methods.