Inheritance & Polymorphism

Task 1:

Consider a class *BankAccount* that has

Two attributes i.e. *accountID* and *balance* and A function named *balanceInquiry()* to get information about the current amount in the account

Derive two classes from the *BankAccount* class i.e. *CurrentAccount* and the *SavingsAccount*. Both classes (*CurrentAccount* and *SavingsAccount*) inherit all attributes/behaviors from the *BankAccount* class. In addition, followings are required to be the part of both classes

Appropriate *constructors* to initialize data fields of base class

A function named *amountWithdrawn(amount)* to withdraw certain amount while taken into account the following conditions

While withdrawing from *current account*, the minimum balance should not decrease *Rs. 5000* o While withdrawing from *savings account*, the minimum balance should not decrease *Rs. 10,000*

amountDeposit(amount) to deposit amount in the account

In the *main()* function, create instances of derived classes (i.e. *CurrentAccount* and *SavingsAccount*) and invoke their respective functions to test their working.

Task 2:

Consider a base class named *Employee* and its derived classes *HourlyEmployee* and *PermanentEmployee* while taking into account the following criteria.

Employee class has two data fields i.e. a *name* (of type string) and specific *empID* (of type integer)

Both classes (*HourlyEmployee* and *PermanentEmployee*) have an attribute named *hourlyIncome*

Both classes (*HourlyEmployee* and *PermanentEmployee*) have *three-argument* constructor to initialize the *hourlyIncome* as well as data fields of the base class

Class *HourlyEmployee* has a function named *calculate_the_hourly_income* to calculate the income of an employee for the actual number of hours he or she worked. One hour income is *Rs. 150*

Similarly, *PermanentEmployee* class has function named *calculate_the_income* to calculate the income of an employee that gets paid the salary for exact *240* hours, no matter how many actual hours he or she worked. Again, one hour salary is *Rs. 150*.

Implement all class definitions with their respective *constructors* to initialize all data members and functions to compute the total income of an employee. In the *main()* function, create an instance of both classes (i.e. *HourlyEmployee* and *PermanentEmployee*) and test the working of functions that calculate total income of an employee.

Task 3:

Consider a class *Computer* having Two fields (i.e. *companyName*, *price*) and A single function named *show()*

A class named *Desktop* inherits *Computer* class and adds fields representing *color, monitor size,* and *processor type* and Override function named *show()* to display values of its all attributes

A class named *Laptop* inherits *Computer* class and adds fields representing *color, size, weight,* and *processor type* and Override function named *show()* to display values of its all attributes

In *Main()* instantiate objects of derived classes to access respective *show()* functions to see the polymorphic behavior.

Task 4:

Create a *payroll* system in which two types of employees are paid weekly.

- *Salaried employees:* fixed salary irrespective of hours.
- *Commission employees:* paid by a percentage of sales.

The information known about each employee is his/her first name, last name and national identity card number. The rest depends on the type of employee.

Step 1: Define Employee Class

Being the base class, *the Employee* class contains the common behavior.
Add *first Name*, *last Nam*e and *CNICNumber* as attributes of type String.

Provide *getters* and *setters* for each attribute.

Write *default* and *parameterized* constructors. Define *earnings ()* method as shown below

public double earnings () {return 0.00;}

Step 2: Define Salaried Employee Class Extend this class form *Employee* class.

Add *weeklySalary* as an attribute of type double. Provide *getter* and *setters* for this attribute. Make sure that weeklySalary never sets to negative value.

Write *default* and *parameterized* constructor. Don't forget to call default and parameterized constructors of Employee class.

Override earnings () method to implement class specific behavior as shown below

public double earnings () {return weeklySalary;}

Step 3: Define Commission Employee Class

Extend this class form *Employee* class.

Add *grossSales* and *commissionRate* as attributes of type double

Provide *getters* and *setters* for these attributes. Make sure that grossSales and commissionRate never set to a negative value.

Write *default* and *parameterized* constructor. Don't forget to call default and parameterized constructors of Employee class.

Override earnings () method to implement class specific behavior as shown below
public

double earnings () {return grossSales * commisionRate;}

Write a Test_Main class that asks the user to input the type of Employee to create objects of particular type based on input from the user. For this you have to show a simple option on the terminal and make the decision based on the option selected. Via selected objects of Employee print employee information and earnings.