Object-Oriented Programming Lab#4

Today's Topics

- Class and Object
- Initialization of fields.
- Constructor
- String concatenation

Problems/Assignments

Problem#1

Create a **Banking** System, where a user can create **new account, deposit money, withdraw money and check the balance** of his/her account.

What you need to do:

1) Create a **BankAccount** class which has **3 instance variables**; ***name, accNum*** and ***balance***.

   a. Create a ***constructor*** with 3 arguments and initialize the attributes.

   The class also has the **following 4 methods**;

   b. void deposit(double depAmount)
   - Inside the method the ***balance*** variable needs to be increased by the "***depAmount***" amount.

   c. void withdraw(double withAmount)
   - The ***balance*** is decreased by "***withAmount***" amount. We have to make sure the ***balance*** does not become negative.

   d. double getBalance()
   - The method returns the ***balance***.

   e. void display()
   - This method displays the attributes in the format "Name:[name]; AccNum:[accNum]; Balance:[balance]". Use ***toString***() method to get the formatted string.

2) Now create an **application class** (that has the main method) named "**Bank**" which will have the **main** method.
   - In the main method, you need ***to create an account*** first. So, take input for the 3 fields (name, accNum, balance) from the user. After taking the input, create a ***BankAccount*** object.
   - After creating the account, you have to provide a **menu** on the console. It will take user input to decide what action to take.
     - Input '1' means **deposit** money. For this input, you have to ask user for the amount of money he wants to deposit.

- Input ‘2’ means **withdraw** money. So, you have to ask user for the amount of money he wants to withdraw. Also you need to prompt if he needs to know the balance. If yes, you need to display the balance before the withdrawal and after the withdrawal.

- Input ‘3’ means **display** the **balance** of the account. In that case you have to display the balance.

- Input ‘0’ means **exit** the system.

Problem#2:

Create an Inventory management system for “UAP Bazar” online store. For simplicity we will work with one product today. Each Product is identified by **its name, id, category and price**. The System should be able **to keep track of the product, check the price, update the price and view the product** info.

What you need to do:

1) Create a **Product** class which has **4 instance variables*; name, id, category*** *and* ***price***.

   a. Create a constructor that takes initial value for those 4 attributes and initializes those attributes.

   Create the **following 4 methods as** described

   a. void updatePrice(double newPrice)

   - Inside the method the ***price*** attributes need to be set to this ***newPrice***.

   b. double getPrice()

   - The method returns the ***price***.

   c. double getDiscountedPrice(double discountPercentage)

   - Store sometimes provide 10-30% discount on certain products. The method will return the ***price*** after discount.

   d. void display()

   - This method displays the attributes.

2) Now create another class **UapBazar** and implement the **main** method. In main method do the following.
   a. In the main method, you need ***to create an object of Product*** first. So, take input for the 4 fields/attributes from the user. After taking the input, create a ***Product*** object.
   b. After creating the object, you have to provide a **menu** on the console. It will take user input to decide what action to take.

      i. Input '1' means **update price**. For this input, you have to ask user for the updated price and call appropriate method to update the price.

      ii. Input '2' means **get discounted price**. So, you have to ask user for the discount amount. After getting the discount amount call appropriate method.

      iii. Input '3' means **display** the **product** info. In that case you have to display the product details.

      iv. Input '0' means **exit** the system.

Problem#3:

Create an Employee Record System for "UAP HR" department. For simplicity we will work with one employee today. Each Employee is identified by **his/her name, employee id and position/designation**. Each employee is paid a fixed monthly **salary** regardless of the number of hours he/she worked. The System should be able **to check the salary of an employee, update the salary and view the employee** info.

What you need to do:

1) Create an **Employee** class which has **4 instance variables*; name, id, designation*** *and* ***salary***.
   a. Create a constructor that takes initial value for those 4 attributes and initializes those attributes.

   Create the **following 3 methods as** described
   e. void updateSalary(double newSal)
   - Inside the method the ***salary*** attributes need to be set to this ***newSal***.
   f. double getSalary()
   - The method returns the ***salary***.
   g. void display()
   - This method displays the attributes in the format "Name:[name]; Id:[id]; Designation:[designation]; Salary:[salary]".

2) Now create another class **UapHr** and implement the **main** method. In main method do the following.
   a. In the main method, you need ***to create an object of Employee*** first. So, take input for the 4 fields/attributes from the user. After taking the input, create an ***Employee*** object.
   b. After creating the employee object, you have to provide a **menu** on the console. It will take user input to decide what action to take.

      i. Input ‘1’ means **update salary**. For this input, you have to ask user for the updated salary and call appropriate method to update the salary.

      ii. Input ‘2’ means **get salary**. So, call appropriate method to get the salary and print the salary.

      iii. Input ‘3’ means **display** the **Employee** info. In that case you have to display the employee details.

      iv. Input ‘0’ means **exit** the system.