Object-Oriented Programming Lab#4

Today's Topics

- Class and Object
- Initialization of fields.
- Constructor
- String concatenation
- Array of reference type

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:

Update **Problem#1** to handle **multiple accounts**, where user can do the following

- create more than one account
- deposit money to any account
- withdraw money from any account
- Get the balance info of any account.

What you need to do:

In the main method, create an **array/ArrayList** of **BankAccount** type. And do the following

1) Add **2 more options** in your previous menu for creating new account and displaying the account info.
    a. When user chooses create account option, take input from user for the 3 fields, create a ***BankAccount*** Object and add to the ***BankAccount*** **array/ArrayList**.
    b. If user chooses display option, access each element of the **BankAccount** array and call the **display()** method.
2) For the **deposit**, **withdraw** and **getBalance** options, now you have to take the account number as input and do the following.
    a. **Find** the account from the Array
    b. **Call** the respective method with appropriate parameter.

Problem#3:

Update the **Online Store of Lab#3** to handle **multiple products**, where user can do the following

- Add new product to the system
- Update the price of a product
- Get the discounted price of a product
- View the list of products with their details.

Problem#4:

Update the **Employee Management System of Lab#3** to handle **multiple employees**, where user can do the following

- Add new employee to the system
- Update the salary of an Employee
- Get the salary of an Employee
- View the list of employees with their details.