

Object-Oriented Programming Lab#4

Today's Topics

- Class and Object
- Constructor
- String concatenation
- JOptionPane
- Menu

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 and add the following.
 - a. Add **3 instance variables**; *name*, *id* and *balance*.
 - b. Create a constructor which will take parameters for all 3 attributes. Inside the constructor, initialize the attributes with that parameter passed to the constructor.
 - c. Add the **following 4 methods** as described;
 - i. **public void deposit(double depAmount)**
Inside the method increase the *balance* by the "*depAmount*" amount.
 - ii. **public void withdraw(double withAmount)**
Inside the method decrease the *balance* by "*withAmount*" amount. **Do necessary checks so that the balance does not become negative.**
 - iii. **public double getBalance()**
Return the *balance* attribute from the method.
 - iv. **public void display()**
Inside the method, display the attributes in the format "Name:[name]; Id:[id]; Balance:[balance]".
- 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 '4' means **display** the **details** of the account. In that case you have to display the details (name, id, 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. **public void updatePrice(double newPrice)**
 - Inside the method the **price** attributes need to be set to this **newPrice**.
 - b. **public double getPrice()**
 - The method returns the **price**.
 - c. **public double getDiscountedPrice(double discountPercentage)**
 - Store sometimes provide 10-30% discount on certain products. The method will return the **payable amount** after discount.
 - d. **public 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 '4' means **display** the **details** of the product. In that case you have to display the details (name, id, category, price).
 - v. Input '0' means **exit** the system.

Problem#3:

Create a Student Record System for "UAP CSE" department. For simplicity we will work with one Student today. Each Student is identified by **his/her name, id, creditCompleted, and cgpa**. The System should be able **to check the CGPA of a student, update the cgpa and view the student** info. For today's lab we will define the **Student** class, create object and call some methods.

What you need to do:

- 1) Create a **Student** class and add the following inside the class.
 - a. Add **4 instance variables; name, id, creditCompleted and cgpa**.
 - b. Create a constructor which will take parameters for all 3 attributes. Inside the constructor, initialize the attributes with that parameter passed to the constructor.
 - c. Add the **following 2 methods** as described
 - i. **public void updateCgpa(double courseCredit, double courseGpa)**
This method is to implement the cgpa changes when a Student pass a course such as CSE 201. When a student passes a course, his/her cgpa get updated and also the total credit he/she completed. To mimic this scenario, inside the method, calculate the

new **cgpa** based on the **creditCompleted**, **courseCredit** and **courseGpa**. Also update the **creditCompleted**.

ii. **public void display()**

This method displays the attributes in the format "Name:[name]; Id:[id]; Credit Completed:[creditCompleted]; CGPA:[cgpa]".

1) Now create another class **UapCSE** and implement the **main** method. In main method do the following.

- a. In the main method, you need **to create an object of Student** class first. So, take input for the 4 fields/attributes from the user. After taking the input, create a **Student** object.
- b. After creating the Student 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 cgpa**. For this input, you have to ask user for the credit of the course and gpa, and then call appropriate method to update the salary.
 - ii. Input '2' means **show cgpa**. So, print the cgpa of the Student.
 - iii. Input '3' means **display** the **Student** info. In that case you have to display the student's details.
 - iv. Input '0' means **exit** the system.