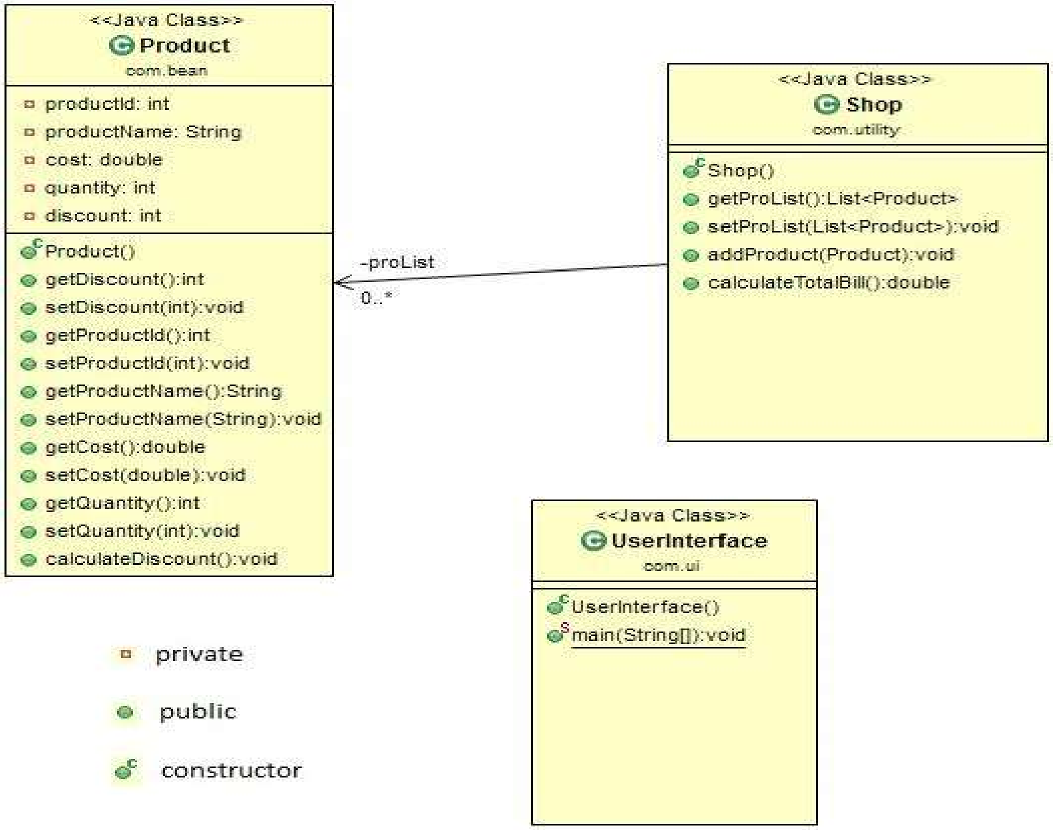
Set 1 Questions

1.Product Bill Calculation

ZEE Shop wants to provide discount for the products purchased by their customers and generate bill. Help ZEE Shop to automate the above process with the help of java.

In the Product class include the given attributes and methods with the access specifiers as specified in the class diagram. The getter methods are used to retrieve the value.



The **calculateDiscount** method should calculate discount based on the total price. The total price needs to be calculated for each product based cost and quantity. The discount should be set to the discount attribute.

totalPrice=unitPrice * quantity

Discount should be calculated based on the below conditions

TotalPrice	Discount Percentage
<500	0
>=500 && <=1000	10
>1000	20

The **addProduct** method of Shop class accepts Product object as argument and adds it to the productList. The product object passed should hold all the values including the discount percentage.

The **calculateTotalBill** method should return the total amount as double based on the cost , quantity and discount of each product in the list. If the list is empty the method should return -1.

[Total Amount= [cost* quantity –(cost*quantity*discount/100)]] of each product in the list and return the sum.

Note: The attribute/method/class name should be specified correctly as given in the class diagram.

Sample Input and Output:

```
Enter the number of products
3
Enter the product details
Enter the product id:
1
Enter the product name:
Furniture
Enter the cost:
500
Enter the quantity:
5
Enter the product details
Enter the product id:
2
Enter the product name:
Eraser
Enter the cost:
5
Enter the quantity:
2

Enter the product details
Enter the product id:
3
```

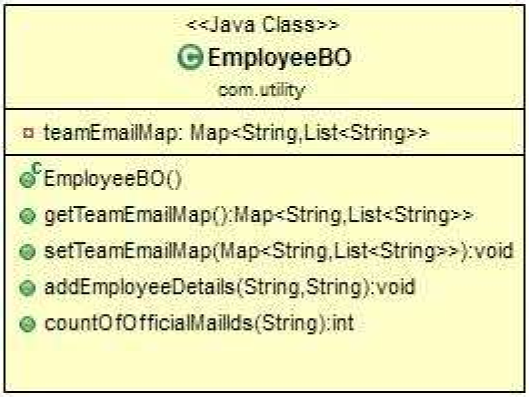
```
Enter the product name:
Bags
Enter the cost:
100
Enter the quantity:
5

Calculated Bill Amount:2460.0
```

2. Team Email Ids

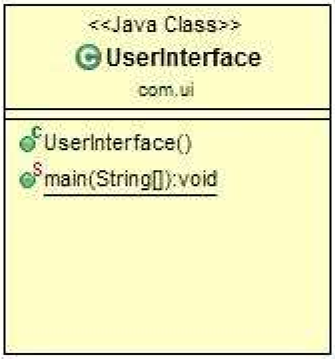
The HR of a leading software company wants to categorize their employee official mail ids based on their team to communicate with employees.

Help them by writing a Java program to store the details of Employees as key-value pair with the team name as key and the list of official mail ids as values.

Consider the class given below:



<<Java Class>>
EmployeeBO
com.utility
teamEmailMap: Map<String,List<String>>
EmployeeBO()
getTeamEmailMap():Map<String,List<String>>
setTeamEmailMap(Map<String,List<String>>):void
addEmployeeDetails(String,String):void
countOfOfficialMailIds(String):int



□ private

● public

●ᶜ constructor

The public void addEmployeeDetails (String teamName,String officialMail) method should add teamName as key and add the officialMail in the list as values in Map. If the teamName(key) is already available in the map then add the officialMail into the list which is already available as value.

If the teamName(key) is not available in the map then add the teamName as key into the map and create a new list as value corresponding to the key and add the officialMail into that list.

The public int countOf OfficialMailIds(String teamName) method should return the number of official Mail ids available in the list corresponding to the teamName (key) passed as argument (String) to this method.

In the Map, if no key matches(case-insensitive) with the teamName which was passed as an argument, then the method has to return -1.

Note: teamName is case-insensitive. Use appropriate collection to maintain insertion order while adding elements into Map and List.

The signature of the above methods are given as part of code skeleton, do not change the method signature.

Sample keys and values of Vechicle Details:

Key(String)	**Value (List<String>)**
Corporate	[john@leadingsoft.com,joyson@leadingsoft.com]
HR	[jacob@leadingsoft.com]
Admin	[james@leadingsoft.com,cristy@leadingsoft.com]

In main class create a main method with the menu as described in sample Input and Output. When user selects the option 1. Add Employee Details, it should get the teamName (String) and OfficialMail (String) from the user and invoke the method addEmployeeDetails (String teamName,String OfficialMail).

When user selects the option 2. Number of Official Email, it should get the teamName (String) as input from the user and invoke the method countOfOfficialMailIds(String teamName) and display the count returned from the method.

When user selects the option 3. Exit, display the message "End" and terminate the program.

Sample Input and Output 1:

1. Add Employee Details

2. Number of Official Email

```
3. Exit
Enter your choice
1
Enter the Team Name
Corporate
Enter the official mail id
john@leadingsoft.com
1. Add Employee Details
2. Number of Official Email
3. Exit
Enter your choice
1
Enter the Team Name
HR
Enter the official mail id
jacob@leadingsoft.com
1. Add Employee Details
2. Number of Official Email
3. Exit
Enter your choice
1
Enter the Team Name
corporate  // note teamName is case-Insensitive
Enter the official mail id
joyson@leadingsoft.com
1. Add Employee Details
2. Number of Official Email
3. Exit
Enter your choice
1
Enter the Team Name
Admin
Enter the official mail id
```

```
james@leadingsoft.com
1. Add Employee Details
2. Number of Official Email
3. Exit
Enter your choice
1
Enter the Team Name
Admin
Enter the official mail id
cristy@leadingsoft.com
1. Add Employee Details
2. Number of Official Email
3. Exit
Enter your choice
2
Enter the team Name to find number of official mail
corporate // note teamName is case-Insensitive
2
1. Add Employee Details
2. Number of Official Email
3. Exit
Enter your choice
3
End
```

3. HACKERS

Maya went for a java coding contest and she has been asked to provide the solution for the following scenario, She has to get the full name **[Firstname Lastname]** as the input from the user and she has to find the first non-repeating character from the Firstname and last non-repeating character from the Lastname. If Firstname/Lastname/both does not contain non-repeating characters display **“No non-repeating characters”**. Help her to win the contest.

Assume the name consist of firstname space lastname in uppercase.

Note:

If the length of firstname is less than 4, then display “<<first name>> is too short”.

If the length of lastname is less than 4, then display “<<last name>> is too short”.

If the length of firstname & lastname is less than 4, then display “<<firstname lastname>> is too short”.

Sample Input 1:

Enter the firstname and lastname

MAYA YASHWANTH

Sample Output1:

First non-repeating character in MAYA

M

Last non-repeating character in YASHWANTH

T

Sample Input 2:

Enter the firstname and lastname

AYYA SWAMY

Sample Output2:

No non-repeating characters in AYYA

Last non-repeating character in SWAMY

Y

Sample Input 3:

Enter the firstname and lastname

JOHN AZZA

Sample Output3:

First non-repeating character in JOHN

J

No non-repeating characters in AZZA

Sample Input 4:

Enter the firstname and lastname

BARBARA AZZA

Sample Output4:

No non-repeating characters in BARBARA

No non-repeating characters in AZZA

Sample Input 5:

Enter the firstname and lastname

ABI SAM

Sample Output5:

ABI SAM is too short

Sample Input6:

Enter the firstname and lastname

SAMUVEL RIO

Sample Output6:

RIO is too short

4. **Email Validator**

Mary owns a departmental store. She is planning to collect informations of her customer such as customeName and customerEmailId. Since she wants to send offer details to their customers through mail she wants to check the mail id given by customer is correct by confirming the mail id once again before registration.

If the customer EmailId and confirm Emailid is same the system should display "Registered Sucessfully". If the customer EmailId and confirm mailid is different the system should alert her with the message “Customer Email Id does not match”. So help her by writing a Java program which will throw a user defined exception “MailMismatchException”.

Partial Code is given to do the above task. In the Validator class, you are provided with the method

public static boolean validateEmail(customerEmailId,confirmMail).Logic for validating the eMail is already given as part of the partial code skeleton. Complete the rest of the code as below.

If the customerEmailId matches (case-insensitive) exactly with confirmMail ,this method returns true.

If the customerEMailId doesnot matches (case-insensitive) with confirmMail this method should throw a “MailMismatchException”.

From the main method invoke the validateEmail method and handle the exception and print the appropriate message.

Sample Input and Output 1:

Enter the customer Name

Jaison

Enter the customer Email Id

jaison2154@yahma.co.in

Re-enter the customer Email Id to confirm

jaison2154@yahma.co.in

Registered Sucessfully

Sample Input and Output 2:

Enter the customer Name

Jaison

Enter the customer Email Id

jaison2154@yahma.co.in

Re-enter the customer Email Id to confirm

jaison.123@yahma.co.in

Customer Email Id does not match