Object Oriented Programming III

Q1:-

**Address Entity**

Create a class named Address with the following attributes:

* doorNumber of type Integer
* street of type String
* city of type String
* district of type String
* state of type String
* pinCode of type Integer

Include appropriate getters, setters and constructors.

Include a method that lists all districts in a particular state sorted in alphabetical order. If the state is not present in the list of addresses, print “No records found”.

Include a method that lists all cities in a particular district sorted in alphabetical order. If the district is not present in the list of addresses, print “No records found”.  
  
**Use appropriate collections.**

Write a main method to test the above class.

**Input and Output Format:**

Refer sample input and output.

All text in bold corresponds to input and the rest corresponds to output.

**Sample Input and Output:**

Enter the number of addresses to be inserted

**2**

Enter address 1 details

Enter door number

**2**

Enter street name

**LMC**

Enter city name

**Ooty**

Enter district name

**Nilgiris**

Enter state name

**Tamilnadu**

Enter pin code

**641044**

Enter address 2 details

Enter door number

**5**

Enter street name

**Red Cross**

Enter city name

**Satyamangalam**

Enter district name

**Erode**

Enter state name

**Tamilnadu**

Enter pin code

**642054**

Enter a state name

**Tamilnadu**

List of districts in Tamilnadu are

Erode

Nilgiris

Enter a district name

**Erode**

List of cities in Erode are

Sathyamangalam

Q2:-

**User Entity  
[Prerequisite for this problem :  Address Entity]**

Create a class named User with the following attributes

name of type String

mobileNumber of type String

email of type String

address of type Address

creditCardNumber of type String

Include appropriate getters, setters and constructors.

Include methods to

1. Search for user details using mobile number. If no records exist, print “No records found”.
2. Search for user details using mobile number and then change the email id of the above user to new id provided. If no records exist, print “No records found”.
3. List the names of users in which the given string is a substring. List the users sorted in alphabetical order. If no records exist, print “No records found”.
4. List the names of users from a particular district. List the users sorted in alphabetical order. If no records exist, print “No records found”.

Write a main method to test the above class.

**Input and Output Format:**

Refer sample input and output.

All text in bold corresponds to input and the rest corresponds to output.

**Sample Input and Output:**

Enter the number of user details to be inserted

**2**

Enter details of user 1

Enter name

**Ram**

Enter mobile number

**9987612345**

Enter email address

**ram.am@amphisoft.in**

Enter address details

Enter door number

**2**

Enter street name

**LMC**

Enter city name

**Ooty**

Enter district name

**Nilgiris**

Enter state name

**Tamilnadu**

Enter pin code

**641044**

Enter credit card number

**FG5423**

Enter details of user 2

Enter name

**Sita**

Enter mobile number

**9237668345**

Enter email address

**sita.am@amphisoft.in**

Enter address details

Enter door number

**2**

Enter street name

**Red Cross**

Enter city name

**Sathyamangalam**

Enter district name

**Erode**

Enter state name

**Tamilnadu**

Enter pin code

**641354**

Enter credit card number

**AG1234**

Menu

1) Search for user details using mobile number.

2) Search for user details using mobile number and then change the email id of the above user to new id provided.

3) List the names of users in which the given string is a substring.

4) List the names of users from a particular district.

Any other choice : Exit

Enter your choice

**1**

Enter the mobile number

**9894049708**

No records found

Enter your choice

**1**

Enter the mobile number

**9237668345**

User Details:

Name : Sita

Email : sita.am@amphisoft.in

Credit Card Number : AG1234

Enter your choice

**2**

Enter the mobile number

**9237668345**

Enter the new email id

**sita.amph@amphisoft.in**

Enter your choice

**1**

Enter the mobile number

**9237668345**

User Details:

Name : Sita

Email : sita.amph@amphisoft.in

Credit Card Number : AG1234

Enter your choice

**3**

Enter the substring

**ta**

List of user names that contain the substring ta

Sita

Enter your choice

**4**

Enter the district name

**Nilgiris**

List of users from Nilgiris

Ram

Enter your choice

**5**

Application Terminated

Q3:-

**Author**

Create a class named Author with the following private member variables.

* name of type String
* email of type String
* gender of type String

The class Author implements Comparable interface.

Include appropriate getters and setters.

Include appropriate constructors.

Override the toString () method defined in the Object class to display the author details in the given format:

Ex:

name (gender) contact at email

A (Male) contact at A@gmail.com

Override the compareTo() method defined in the Comparable interface to compare 2 Author objects based on author name.

Create another class named AuthorBO. There are no member variables in this class.

Include the following public methods in this class

|  |  |
| --- | --- |
| Method prototype | Method description |
| public void displayAllAuthorDetails (List<Author> authorList) | Display the details of all authors in the specified format using the code snippet given below. |
| public void listAuthorNamesInSortedOrder (List<Author> authorList) | Prints the line as shown in the output and displays the names of authors sorted lexicographically. |
| public Author findAuthorByName (List<Author> authorList, String name) | Returns the author object where the author name is 'name'. If no such object is found, this method returns null. |

Use the following code snippet to format the display of authors.

System.out.format("%-20s %-20s %-20s\n", "Name","Email","Gender");   // Java  
or  
Console.WriteLine("{0,-20} {1,-20} {2,-20}","Name","Email","Gender");  // C#

Create another class and write a main method to test the above classes.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Sample Input and Output 1:**

Enter the author name

**Chetan Bhagat**

Enter the author email id

**chetan.bhagat@gmail.com**

Enter the author's gender

**Male**

Do you want to add the details of another author? Type Yes/No (Not case sensitive)

**yes**

Enter the author name

**Arundati Roy**

Enter the author email id

**a.roy@gmail.com**

Enter the author's gender

**Female**

Do you want to add the details of another author? Type Yes/No (Not case sensitive)

**no**

Name Email Gender

Chetan Bhagat chetan.bhagat@gmail.com Male

Arundati Roy a.roy@gmail.com Female

Enter the name of the author to be searched

**Arundati**

Author not found

Alphabetically sorted author names

Arundati Roy

Chetan Bhagat

**Sample Input and Output 2:**

Enter the author name

**Chetan Bhagat**

Enter the author email id

**chetan.bhagat@gmail.com**

Enter the author's gender

**Male**

Do you want to add the details of another author? Type Yes/No (Not case sensitive)

**yes**

Enter the author name

**Arundati Roy**

Enter the author email id

**a.roy@gmail.com**

Enter the author's gender

**Female**

Do you want to add the details of another author? Type Yes/No (Not case sensitive)

**no**

Name Email Gender

Chetan Bhagat chetan.bhagat@gmail.com Male

Arundati Roy a.roy@gmail.com Female

Enter the name of the author to be searched

**Arundati Roy**

Arundati Roy (Female) contact at a.roy@gmail.com

Alphabetically sorted author names

Arundati Roy

Chetan Bhagat