**Question:**

**Design a class : Movie**

**add attributes as private:**

**movieName String**

**yearRelease int**

**director String**

**movieType String**

**Add methods as :**

**setter/getter**

**constructor (default & parameterized )**

**override toString() as**

**String.format("%-20s %-20s %-20s %-10s",movieName,director,movieType,yearRelease);**

**Create a Main class to make following Input/Output UI:**

**How many Movie?**

**4**

**Sholay,1900,R.Sippy,Action**

**Shola,1902,Sippy,Action**

**Herapheri,1988,Raj,Comedy**

**Kalank,2019,R.Sippy,Romance**

**Movie Director Type Year**

**Sholay R.Sippy Action 1900**

**…….**

**Q-1.1.**

**Design a class : Employee**

**firstName String**

**lastName String**

**gender String**

**dtOfJoin Date**

**1. add setter/getter**

**2. add constructor (default & parameterized )**

**3. override toString() as**

**String.format("%-25s %-10s %-10s",firstName+" "+lastName,gender,dtOfJoin);**

**4. Take some employee information like:**

**How many Employee?**

**4**

**Ram,Das,Male,01-12-1980**

**Rama,Das,Female,01-12-1998**

**Ranjan,Sen,Male,01-11-1980**

**Raja,Das,Male,01-12-1980**

**Show them on console:**

**Name Gender Joining**

**Ram Das Male 01-12-1980**

**Rama Das Female 01-12-1998**

**....**

**Question 1:**  
  
  
a)Create a Class Contact with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| name | String |
| company | String |
| title | String |
| mobile | String |
| alternateMobile | String |
| email | String |
| dateCreated | java.util.Date |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Contact(String name, String company, String title, String mobile, String alternateMobile, String email, Date dateCreated)**  
  
b)Create a Class Group with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| name | String |
| contactList | List<Contact> |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Group(String name, List<Contact> contactList).** In constructor pass the contactList value as an empty list. Only one group will be present at a time.  
  
c) Create the following static method in Contact class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public static Contact createContact(String detail) | This method accepts a string which contains contact details separated by commas. Split the details and create a contact object from the details and return the contact object. |

The contact details should be given as a comma-separated value in the below order,  
**name,company,title, mobile, alternateMobile, email, dateCreated**  
  
d) Create the following methods in Group class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public void addContactToGroup(Contact contact) | This method accepts a contact object and add the contact to the contact list of the current group. |
| public Boolean removeContactFromGroup(String name) | This method will get a name of the contact and delete the contact with the specified name from the current group. If a contact with the given contact name found, delete the contact and return **true**. If a contact with the contact name is not found return **false**. |
| public void displayContacts() | This method will display the contact list in the current group. If the contact list is empty display "**No contact to show"**, else display "Contacts in [group name]" and display all the contact details in the specified format. Where [group name] specifies the name of the group. |

After deletion, if true is returned print "**Contact successfully deleted**", else print "**Contact not found in the group**".  
  
**Note:** The above print statements should be present in the main method.  
  
When the “contact” object is printed, it should display the following format  
Print format:  
**String.format("%-15s %-15s %-20s %-15s %-15s %-15s %s", "Name","Company","Title","Mobile","Alternate Mobile","Email","Date Created");**  
  
**Sample Input and Output:**  
  
Enter the group name:  
**Activity Group**  
1.Add Contact  
2.Delete Contact  
3.Display contacts  
4.Exit  
Enter your choice:  
**3**  
No contact to show  
1.Add Contact  
2.Delete Contact  
3.Display contacts  
4.Exit  
Enter your choice:  
**1**  
Enter the number of contacts:  
**3  
John,Microsoft,Product Engineer,9876543210,7894651320,john@micro.com,12-01-2018  
Peter,Apple,Team Lead,9654873210,9873216540,peter@apple.com,15-01-2018  
Starc,IBM,HR,9654873210,8973216540,starc@ibm.com,14-01-2018**  
1.Add Contact  
2.Delete Contact  
3.Display contacts  
4.Exit  
Enter your choice:  
**3**  
Contacts in Activity Group

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Company | Title | Mobile | Alternate Mobile | Email | Date Created |
| John | Microsoft | Product Engineer | 9876543210 | 7894651320 | john@micro.com | 12-01-2018 |
| Peter | Apple | Team Lead | 9654873210 | 9873216540 | peter@apple.com | 15-01-2018 |
| Starc | IBM | HR | 9654873210 | 8973216540 | starc@ibm.com | 14-01-2018 |

1.Add Contact  
2.Delete Contact  
3.Display contacts  
4.Exit  
Enter your choice:  
**2**  
Enter the name of the contact to be deleted:  
**Linda**  
Contact not found in the group  
1.Add Contact  
2.Delete Contact  
3.Display contacts  
4.Exit  
Enter your choice:  
**2**  
Enter the name of the contact to be deleted:  
**Starc**  
Contact successfully deleted  
1.Add Contact  
2.Delete Contact  
3.Display contacts  
4.Exit  
Enter your choice:  
**1**  
Enter the number of contacts:  
**1  
Mark,IBM,HR,9654873210,8973216540,mark@ibm.com,14-01-2018**  
1.Add Contact  
2.Delete Contact  
3.Display contacts  
4.Exit  
Enter your choice:  
**3**  
Contacts in Activity Group

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Company | Title | Mobile | Alternate Mobile | Email | Date Created |
| John | Microsoft | Product Engineer | 9876543210 | 7894651320 | john@micro.com | 12-01-2018 |
| Peter | Apple | Team Lead | 9654873210 | 9873216540 | peter@apple.com | 15-01-2018 |
| Mark | IBM | HR | 9654873210 | 8973216540 | mark@ibm.com | 14-01-2018 |

1.Add Contact  
2.Delete Contact  
3.Display contacts  
4.Exit  
Enter your choice:  
**4**

**Question 2:**  
  
  
a) Create a Class **Contact** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| name | String |
| company | String |
| title | String |
| mobile | String |
| alternateMobile | String |
| email | String |
| dateCreated | java.util.Date |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: Contact(String name, String company, String title, String mobile, String alternateMobile, String email, Date dateCreated)  
  
b) Create a class **ContactBO**with the following methods,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public List<Contact> findContact (List<Contact> contactList,List<String> name) | This method accepts a list of contacts and list of names as arguments and returns a list of contacts with the given names. |
| public List<Contact> findContact (List<Contact> contactList,Date dateCreated) | This method accepts a list of contacts and created date as arguments and returns a list of contacts that were created on the given specified date. |
| public List<Contact> findContact (List<Contact> contactList,String emailDomain) | This method accepts a list of contacts and a domain name as arguments, then find all the contacts with the given domain from the contact list and return the list of contacts with the specified email domain name. |

The contact details should be given as a comma-separated value in the below order,  
name,company,title, mobile, alternateMobile, email, dateCreated  
  
Get the number of contacts and the contact details, build a contact list and perform search by name, date created and email domain.  
  
When the “contact” object is printed, it should display the following details  
Print format:  
**System.out.format("%-15s %-15s %-20s %-15s %-20s %-15s %s\n", "Name","Company","Title","Mobile","Alternate Mobile","Email","Date Created");**  
  
**Note:** The contact lists are displayed in the main method.  
             If any other choice is selected, display "**Invalid Choice**"  
  
**Sample Input and Output 1:**  
  
Enter the number of contact details:  
**5  
John,Microsoft,Product Engineer,9876543210,7894651320,john@micro.com,12-01-2018  
Peter,Apple,Team Lead,9654873210,9873216540,peter@apple.com,15-01-2018  
Starc,IBM,HR,9654373210,8989216540,starc@ibm.com,14-01-2018  
Tony,Intel,HR,9694873210,8973212540,tony@intel.com,14-01-2018  
Mark,Intel,Manager,7994873210,9873212340,mark@intel.com,13-01-2018**  
Enter a search type:  
1.Name  
2.Date created  
3.Email domain  
**1**  
Enter the names:  
**John,Tony**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Company | Title | Mobile | Alternate Mobile | Email | Date Created |
| John | Microsoft | Product Engineer | 9876543210 | 7894651320 | john@micro.com | 12-01-2018 |
| Tony | Intel | HR | 9694873210 | 8973212540 | tony@intel.com | 14-01-2018 |

**Sample Input and Output 2:**  
  
Enter the number of contact details:  
**5  
John,Microsoft,Product Engineer,9876543210,7894651320,john@micro.com,12-01-2018  
Peter,Apple,Team Lead,9654873210,9873216540,peter@apple.com,15-01-2018  
Starc,IBM,HR,9654373210,8989216540,starc@ibm.com,14-01-2018  
Tony,Intel,HR,9694873210,8973212540,tony@intel.com,14-01-2018  
Mark,Intel,Manager,7994873210,9873212340,mark@intel.com,13-01-2018**  
Enter a search type:  
1.Name  
2.Date created  
3.Email domain  
**2**  
Enter the date to search contacts that were created on that date  
**14-01-2018**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Company | Title | Mobile | Alternate Mobile | Email | Date Created |
| Starc | IBM | HR | 9654373210 | 8989216540 | starc@ibm.com | 14-01-2018 |
| Tony | Intel | HR | 9694873210 | 8973212540 | tony@intel.com | 14-01-2018 |

**Sample Input and Output 3:**  
  
Enter the number of contact details:  
**5  
John,Microsoft,Product Engineer,9876543210,7894651320,john@micro.com,12-01-2018  
Peter,Apple,Team Lead,9654873210,9873216540,peter@apple.com,15-01-2018  
Starc,IBM,HR,9654373210,8989216540,starc@ibm.com,14-01-2018  
Tony,Intel,HR,9694873210,8973212540,tony@intel.com,14-01-2018  
Mark,Intel,Manager,7994873210,9873212340,mark@intel.com,13-01-2018**  
Enter a search type:  
1.Name  
2.Date created  
3.Email domain  
**3**  
Enter the Email domain to search contacts that have same email domain  
**intel**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Company | Title | Mobile | Alternate Mobile | Email | Date Created |
| Tony | Intel | HR | 9694873210 | 8973212540 | tony@intel.com | 14-01-2018 |
| Mark | Intel | Manager | 7994873210 | 9873212340 | mark@intel.com | 13-01-2018 |

**Sample Input and Output 4:**  
  
Enter the number of contact details:  
**5  
John,Microsoft,Product Engineer,9876543210,7894651320,john@micro.com,12-01-2018  
Peter,Apple,Team Lead,9654873210,9873216540,peter@apple.com,15-01-2018  
Starc,IBM,HR,9654373210,8989216540,starc@ibm.com,14-01-2018  
Tony,Intel,HR,9694873210,8973212540,tony@intel.com,14-01-2018  
Mark,Intel,Manager,7994873210,9873212340,mark@intel.com,13-01-2018**  
Enter a search type:  
1.Name  
2.Date created  
3.Email domain  
**4**  
Invalid choice

**Question 3:**  
a)Create a Class **Mail** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| Id | Long |
| From | String |
| To | String |
| Subject | String |
| Content | String |
| receivedDate | java.util.Date |
| size | Double |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order:  **public Mail(Long id, String from, String to, String subject, String content, Date receivedDate, Double size)**  
  
b)Create a Class **MailFolder** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| name | String |
| mailList | ArrayList<Mail> |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **MailFolder(String name, List<Mail>mailList).** In constructor pass the mailList value as an empty list. Only one folder will be present at a time.  
  
c) Create the following static method in Mail class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public static Mail createMail(String detail) | This method accepts a string which contains mail details separated by commas. Split the details and create a mail object from the details and return it. |

The mail details should be given as a comma-separated value in the below order,  
**id,from,to, subject, content, receivedDate, size**  
  
d) Create the following methods in MailFolder class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public void addMailToFolder(Mail mail) | This method accepts a mail object and add the mail to the mail list of the current mail folder. |
| public Boolean removeMailFromFolder(Long id) | This method will get the id of the mail and delete the mail with the specified id from the current folder. If a mail with the given id found, delete the mail and return **true**. If a mail with the id is not found return **false**. |
| public void displayMails() | This method will display the mail list in the current folder. If the mail list is empty display "**No mails to show"**, else display "Mails in [folder name]" and display all the mail details in the specified format. Where [folder name] specifies the name of the folder. |

After deletion, if true is returned print "**Mail successfully deleted**", else print "**Mail not found in the folder**". After adding mail to the folder, print "**Mail successfully added**".  
  
**Note:** The above print statements should be present in the main method.  
  
When the “mail” object is printed, it should display the following format  
Print format:  
**String.format("%-10s%-15s%-15s%-15s%-20s%-15s%-10s\n", "Id","From","To","Subject","Content","ReceivedDate","Size"); Display 1 digit after decimal point in Double.**  
  
**Sample Input and Output:**  
  
Enter the name of the folder:  
**Inbox**  
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**3**  
No mails to show  
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**1**  
Enter the details of mail in CSV format:  
**12,john@abc.in,jane@abc.in,Hi,Happy New Year,01-01-2018,10**  
Mail successfully added  
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**1**  
Enter the details of mail in CSV format:  
**16,jack@abc.in,jane@abc.in,Hi,Happy Pongal,14-01-2018,15**  
Mail successfully added  
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**3**  
Mails in Inbox  
Id        From           To             Subject        Content             Received Date  Size        
12        john@abc.in    jane@abc.in    Hi             Happy New Year      01-01-2018     10.0        
16        jack@abc.in    jane@abc.in    Hi             Happy Pongal        14-01-2018     15.0        
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**2**  
Enter the id of the mail to be deleted:  
**13**  
Mail not found in the folder  
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**2**  
Enter the id of the mail to be deleted:  
**16**  
Mail successfully deleted  
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**3**  
Mails in Inbox  
Id        From           To             Subject        Content             Received Date  Size        
12        john@abc.in    jane@abc.in    Hi             Happy New Year      01-01-2018     10.0        
1.Add Mail  
2.Delete Mail  
3.Display Mails  
4.Exit  
Enter your choice:  
**4**

**Question 4:**  
  
  
a)Create a Class **Player** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| name | String |
| dateOfBirth | java.util.Date |
| skill | String |
| numberOfMatches | Integer |
| runs | Integer |
| wickets | Integer |
| nationality | String |
| powerRating | Double |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **public Player(String name, Date dateOfBirth, String skill, Integer numberOfMatches, Integer runs, Integer wickets, String nationality, Double powerRating).**   
  
b)Create a Class **Team** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| name | String |
| playerList | List<Player> |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **public Team(String name, List<Player> playerList).** In constructor pass the playerList value as an empty list. Only one team will be present at a time.  
  
c) Create the following static method in Player class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public static Player createPlayer(String detail) | This method accepts a String which contains player details separated by commas. Split the details and create a player object from the details and return it. |

The player details should be given as a comma-separated value in the below order,  
**name,dateOfBirth,skill,numberOfMatches,runs,wickets,nationality,powerRating**  
  
d) Create the following methods in Team class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public void addPlayerToTeam(Player player) | This method accepts a Player object and add the player to the player list of the current team. |
| public Boolean removePlayerFromTeam(String name) | This method will get the name of the player and delete the player with the specified name from the current team. If a player with the given name found, delete the player and return **true**. If a player with the name is not found return **false**. The players name are unique. |
| public void displayPlayers() | This method will display the player list in the current team. If the player list is empty display "**No players to show"**, else display "Players in [team name]" and display all the player details in the specified format. Where [team name] specifies the name of the team. |

After deletion, if true is returned print "**Player successfully deleted**", else print "**Player not found in the team**". After adding a player to the team, print "**Player successfully added**".  
  
**Note:** The above print statements should be present in the main method.  
  
When the “player” object is printed, it should display the following format  
Print format:  
**System.out.printf("%-15s%-15s%-10s%-15s%-10s%-10s%-15s%-10s\n", "Name","Date of birth","Skill","No of matches","Runs","Wickets","Nationality","Rating"). Display 1 digit after decimal point in Double.**  
  
**Sample Input and Output:**  
  
Enter the name of the Team:  
**Royal Challengers Banglore**  
1.Add Player  
2.Delete Player  
3.Display Players  
4.Exit  
Enter your choice:  
**3**  
No players to show  
1.Add Player  
2.Delete Player  
3.Display Players  
4.Exit  
Enter your choice:  
**1**  
Enter the details of player in CSV format:  
**Virat Kohli,05-11-1988,Batsman,149,4418,4,India,4.7**  
Player successfully added  
1.Add Player  
2.Delete Player  
3.Display Players  
4.Exit  
Enter your choice:  
**1**  
Enter the details of player in CSV format:  
**Ab de Villiers,7-02-1984,Batsman,129,3473,0,S Africa,4.7**  
Player successfully added  
1.Add Player  
2.Delete Player  
3.Display Players  
4.Exit  
Enter your choice:  
**1**  
Enter the details of player in CSV format:  
**Bhuvneshwar,05-02-1990,Bowler,90,158,111,India,4.1**  
Player successfully added  
1.Add Player  
2.Delete Player  
3.Display Players  
4.Exit  
Enter your choice:  
**1**  
Enter the details of player in CSV format:  
**Mitchell Stark,30-01-1990,Bowler,27,96,34,Australia,4.1**  
Player successfully added  
1.Add Player  
2.Delete Player  
3.Display Players  
4.Exit  
Enter your choice:  
**2**  
Enter the name of the player to be deleted:  
**MS Dhoni**  
Player not found in the team  
1.Add Player  
2.Delete Player  
3.Display Players  
4.Exit  
Enter your choice:  
**2**  
Enter the name of the player to be deleted:  
**Bhuvneshwar**  
Player successfully deleted  
1.Add Player  
2.Delete Player  
3.Display Players  
4.Exit  
Enter your choice:  
**3**  
Players in:Royal Challengers Banglore  
Name           Date of birth  Skill     No of matches  Runs      Wickets   Nationality    Rating      
Virat Kohli    05-11-1988     Batsman   149            4418      4         India          4.7         
Ab de Villiers 07-02-1984     Batsman   129            3473      0         S Africa       4.7         
Mitchell Stark 30-01-1990     Bowler    27             96        34        Australia      4.1         
1.Add Player  
2.Delete Player  
3.Display Players  
4.Exit  
Enter your choice:  
**4**

**Question 5:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Create a class named as **Customer**, which contains following private variable/ attributes,   |  |  | | --- | --- | | **Member Field name** | **Type** | | id | Long | | name | String | | gender | Character (M/F) | | email | String | | contactNumber | String | | createdOn | Date (time in 24 hrs clock) dd/MM/yyyy HH:mm:ss | | address | Address | |

Mark all the attributes as private

Create / Generate appropriate Getters & Setters.

Add a default constructor and a parameterized constructor to take in all attributes.

Implement the following methods in the Customer class:

|  |  |  |
| --- | --- | --- |
| **No** | **Method Name** | **Method Description** |
| 1 | public Customer findCustomerById(List<Customer> customerList, Long id) | In this method, that takes up an id and returns a matching customer object. If the object is not found then return null. |
| 2 | public List<Customer> findCustomerListByState(List<Customer> customerList, String state) | In this method, given a state as a parameter, print the list of customers who belong to the state. |

Create a class named as **Address**, which contains following private variable/ attributes,

|  |  |
| --- | --- |
| **Member Field name** | **Type** |
| street | String |
| city | String |
| state | String |
| country | String |
| zipCode | Integer |

Include appropriate getters and setters.

Add a default constructor and a parameterized constructor to take in all attributes.

The customer details will be populated in a static block in this class(provide in the template code).  
Refer Sample Input and Output.  
Use the following format for specified output.  
**"%-15s %-20s %-15s %-15s %s\n", "Name", "Email", "City", "Country", "Zipcode"**  
  
**Problem Specification:**  
If the customer with the id is not found then print "**No Customer with that id**" without quotes.  
If none of the customer belongs to the state then print "**No customer belongs that state**" without quotes.  
  
**Sample Input and Output 1:**

Menu

1. Find customer by id

2. Find customer by states

Enter the choice:

**1**

Enter the Id to find customer:

**5**

Customer Name: Tedmond

Gender: M

Email: [tedmond@a.com](mailto:tedmond@a.com)

Contact Number: +88-7844-8854799658

Street: Port Townsend

City: Tacoma

State: Washington

Country: USA

Zip code: 98412

**Sample Input and Output 2:**

Menu

1. Find customer by id

2. Find customer by states

Enter the choice:

**2**

Enter the state:

**Texas**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Email | Contact no | Street | City | Country | Zipcode |
| Aekerman | [aekerman@a.com](mailto:aekerman@a.com) | +78-7485-9555874846 | Avenue | Plano | USA | 75025 |
| Madeleine | [madeleine@a.com](mailto:madeleine@a.com) | +78-9855-7488742136 | Parc St | Lubbock | USA | 79404 |
| Raymond | [raymond@a.com](mailto:raymond@a.com) | +89-7748-8859112478 | Wall Street | Texas City | USA | 77591 |

Question :

Create a class by the name Student with the following private fields.

Field Dataype

name String

address String

course String

age int

Generate the necessary getters and setters for the above fields.

Add a default constructor for the class Add a parameterized constructor with the following signature. public Student (String name, String address, String course, int age) {}

Override the toString() method for printing the student details in the following format "%-30s%-50s%-30s%-6d"

Implement the Comparable interface to sort the Students based on name in ascending order by overriding compareTo method.

Create another class StudentBO with the following methods.

public void addStudent(List<Student> studentList, Student student) {

/\*Add the student to the list \*/

}

public void printAllStudents(List<Student> studentList){

//Print All the students' details

}

Sample Input 1:

Enter number of students

4

Rohit Sharma, Kanpur-Uttar Pradesh, Java & Web Technologies,22

Shivam Patil, Agra-Uttar Pradesh, Dot NET Technologies,23

Preeti Sengupta,Kolkata-West Bengal, Java & Web Technologies,21

Subham Paul, Surat-Gujarat, Java & Web Technologies,22

Menu

1. Add a new student

2. Print all the students

Enter your choice

1

Enter your details

Jaspreet Kaur, Amritsar-Punjab, Automation Testing,22

Details added

Menu

1. Add a new student

2. Print all the students

Enter your choice

1

Enter your details

Manisankar Iyer, Calicut-Kerala, Dot NET Technologies,23

Details added

Menu

1. Add a new student

2. Print all the students

Enter your choice

2

Student Details

SL No. Name Address Course Age

1. Jaspreet Kaur Amritsar-Punjab Automation Testing 22

2. Manisankar Iyer Calicut-Kerala Dot NET Technologies 23

3. Preeti Sengupta Kolkata-West Bengal Java & Web Technologies 21

4. Rohit Sharma Kanpur-Uttar Pradesh Java & Web Technologies 22

5. Shivam Patil Agra-Uttar Pradesh Dot NET Technologies 23

6. Subham Paul Surat-Gujarat Java & Web Technologies 22