

All about you(---RETIRED---)

Grade settings: Maximum grade: 100

Disable external file upload, paste and drop external content: Yes

Based on: [All about you\(---RETIRED---\)](#)

Run: Yes **Evaluate:** Yes

Automatic grade: Yes

All about You spa wants to automate the details of members . They would like to automate the process of extracting members data based on member type. You have been approached to develop a Java program for their requirements.

Component Specification: MemberMain

| Type (Class) | Attributes | Methods |
|-------------------|--|--|
| MemberMain | private Map<String,String> memberMap | Getter and setter method for the attribute is included in the code skeleton. |

Note: **key:** *memberId* and **value:** *memberType* in *memberMap* attribute

Requirement 1: Find the memberType based on the given memberId.

| Type (Class) | Methods | Responsibilities |
|-------------------|---|--|
| MemberMain | public String findMemberTypeOfGivenMemberId (String memberId) | This method accepts memberId as an argument, and if the given memberId is present on the map, it must return the memberType of the given memberId. Else return a string " Invalid member id " <i>Condition: memberId is case-sensitive.</i> |

Requirement 2: Filter the member Id's based on the given memberType

| Type (Class) | Methods | Responsibilities |
|--------------|---------|------------------|
|--------------|---------|------------------|

| | | |
|-------------------|--|--|
| MemberMain | <pre> public List<String> findMemberIdsBasedOnTheMemberType(String memberType) </pre> | <p>This method accepts memberType as an argument, and filters the members based on the given memberType and returns the list of memberId's.</p> <p>Condition: memberId is case-insensitive.</p> |
|-------------------|--|--|

You are provided with the main method as code template, and it is excluded from evaluation.

Note:

- In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user, and the rest of the text represents the output.
- Ensure to follow the object-oriented specifications provided in the question description.
- Ensure to provide the names for the classes, attributes, and methods as specified in the question description.
- Adhere to the code template, if provided.

SAMPLE INPUT/OUTPUT 1

Enter number of records to be added:

5

Enter the details (Member id : Member type):

MEM101:platinum

MEM102:Gold

MEM103:Platinum

MEM104:Silver

MEM105:Diamond

Enter the member id to be searched

MEM102

MEM102 is a Gold member

Enter the member type to be searched

Platinum

Member ids of Platinum members are

MEM101

MEM103

SAMPLE INPUT/OUTPUT 2

Enter number of records to be added:

3

Enter the details (Member id : Member type):

MEM101:Gold

MEM102:Gold

MEM103:Diamond

Enter the member id to be searched

MEM109

Invalid member id

Enter the member type to be searched

gold

Member ids of gold members are

MEM101

MEM102

SAMPLE INPUT/OUTPUT 3

Enter number of records to be added:

3

Enter the details (Member id : Member type):

MEM101:Platinum

MEM102:Silver

MEM103:Diamond

Enter the member id to be searched

MEM103

MEM103 is a Diamond member

Enter the member type to be searched

Gold

No members were found for the given member type

```
1 import java.util.List;
2 import java.util.Map;
3 import java.util.Scanner;
4 import java.util.ArrayList;
5 import java.util.HashMap;
6
7 public class MemberMain {
8
9     private Map<String,String> memberMap = new HashMap<String,String>();
10
11     public Map<String, String> getMemberMap() {
12         return memberMap;
13     }
14     public void setMemberMap(Map<String, String> memberMap) {
15         this.memberMap = memberMap;
16     }
17     public String findMemberTypeOfGivenMemberId(String memberId){
18         //Fill the code here
19         return null;
20     }
21     public List<String> findMemberIdsBasedOnTheMemberType(String memberType) {
22         //Fill the code here
23         return null;
24     }
25
26     public static void main(String args[]) {
27         //Main method is excluded from evaluation. You are free to write your own code or add lines of code to check the correctness of the functionalities.
28         MemberMain c=new MemberMain();
29     }
30 }
31
32
33
34
35
36
37
```

Qualifier Assessment All about y x +

https://cognizant.tekstac.com/mod/vpl/forms/edit.php?id=110144&userid=137159#

File List Save Compile & Run Evaluate Reset Restore Description

File list

- AllAboutYou
 - src
 - MemberMain

MemberMain.java

```
28     return null;
29 }
30
31
32
33 public static void main(String args[]) {
34
35     //Main method is excluded from evaluation. You are free to write your own code or add lines of code to check the correctness of the functionalities.
36
37     MemberMain c=new MemberMain();
38     List<String> list1=new ArrayList<String>();
39     Map<String, String> map=new HashMap<String,String>();
40     Scanner sc=new Scanner(System.in);
41     System.out.println("Enter number of records to be added:");
42     int n=sc.nextInt();
43     sc.nextLine();
44     System.out.println("Enter the details (Member id : Member type):");
45     String [] memberDetails = new String[n];
46     for(int i=0;i<n;i++) {
47         memberDetails[i] = sc.nextLine();
48     }
49
50     for(int i=0;i<memberDetails.length;i++) {
51         String[] a = memberDetails[i].split(":");
52
53         map.put(a[0], a[1]);
54
55         c.setMemberMap(map);
56     }
57     System.out.println("Enter the member id to be searched");
58     String id=sc.next();
59     sc.nextLine();
60
61     String result=c.findMemberTypeOfGivenMemberId(id);
62     if(result.equals("Invalid member id"))
63         System.out.println("Invalid member id");
64     else
```

Qualifier Assessment All about y x +

https://cognizant.tekstac.com/mod/vpl/forms/edit.php?id=110144&userid=137159#

File List Save Compile & Run Evaluate Reset Restore Description

File list

- AllAboutYou
 - src
 - MemberMain

MemberMain.java

```
50     for(int i=0;i<memberDetails.length;i++) {
51         String[] a = memberDetails[i].split(":");
52
53         map.put(a[0], a[1]);
54
55         c.setMemberMap(map);
56     }
57     System.out.println("Enter the member id to be searched");
58     String id=sc.next();
59     sc.nextLine();
60
61     String result=c.findMemberTypeOfGivenMemberId(id);
62     if(result.equals("Invalid member id"))
63         System.out.println("Invalid member id");
64     else
65     {
66         System.out.println(id+" is a "+result+" member");
67     }
68
69     System.out.println("Enter the member type to be searched");
70     String memberType=sc.nextLine();
71     list1=c.findMemberIdsBasedOnTheMemberType(memberType);
72     if(list1.size()==0)
73     {
74         System.out.println("No members were found for the given member type");
75     }
76     else
77     {
78         System.out.println("Member ids of "+memberType+" members are");
79         for(String s:list1)
80         {
81             System.out.println(s);
82         }
83     }
84 }
85 }
86 }
```