

Code Cave(---RETIRED---)

Grade settings: Maximum grade: 100

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

Based on: [Code Cave\(---RETIRED---\)](#)

Run: Yes **Evaluate:** Yes

Automatic grade: Yes

Code Cave is a well-known placement training institute with numerous branches across the United States. They have planned to provide appraisals for the trainers based on the performance ratings. The manager intimates a software developer to help in their process. You being the software developer, develop a Java program based on the requirements.

Component Specification: TrainerMain

Type (Class)	Attributes	Methods
TrainerMain	private Map<String, Float> trainerMap	Getter and setter methods for the attribute are included in the code skeleton.

Note: **key:** *trainerId* **value:** *performanceRating* for *trainerMap* attribute

Requirement 1: Find performance rating of the given trainerId

Type (Class)	Methods	Responsibilities
TrainerMain	public float findPerformanceRatingOfGivenTrainerId (String trainerId)	This method accepts trainerId as an argument. If the trainerId is present on the Map, it must return the performance rating of the given trainerId . Else return -1. <i>condition: trainerId is case-sensitive.</i>

Requirement 2: Find the trainerId's with low rating

Type (Class)	Methods	Responsibilities
TrainerMain	<pre>public List<String> findTrainerIdsWithLowPerformanceRating()</pre>	<p>This method filters the trainers with low performance rating and returns the list of trainerId's that satisfy the below condition.</p> <p><i>Condition: trainerId's with rating less than or equal to 3 is considered as low performance rating.</i></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 trainer records (Trainer id : Performance rating):

COD101:2.4

COD102:4.5

COD103:3.7

COD104:3

COD105:4

Enter the trainer id to be searched

COD103

Performance rating of trainer id COD103 is 3.7

Trainer id of the trainers with low rating are:

COD104

COD101

SAMPLE INPUT/OUTPUT 2

Enter number of records to be added:

3

Enter the trainer records (Trainer id : Performance rating):

COD201:4

COD202:4.5

COD203:3.4

Enter the trainer id to be searched

COD202

Performance rating of trainer id COD202 is 4.5

No trainers have a low rating

SAMPLE INPUT/OUTPUT 3

Enter number of records to be added:

4

Enter the trainer records (Trainer id : Performance rating):

COD201:4

COD202:4.5

COD203:2.4

COD204:2.7

Enter the trainer id to be searched

COD206

COD206 is an invalid trainer id

Trainer id of the trainers with low rating are:

COD203

COD204

```
1 import java.util.*;
2 import java.util.Map.Entry;
3
4 public class TrainerMain {
5     private Map<String, Float> trainerMap = new HashMap<>();
6
7
8
9
10    public Map<String, Float> getTrainerMap() {
11        return trainerMap;
12    }
13
14    public void setTrainerMap(Map<String, Float> trainerMap) {
15        this.trainerMap = trainerMap;
16    }
17
18    public float findPerformanceRatingOfGivenTrainerId(String trainerId)
19    {
20        // Fill the code here
21        return 0;
22    }
23
24    public List<String> findTrainerIdsWithLowPerformanceRating()
25    {
26        // Fill the code here
27
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        TrainerMain trainer=new TrainerMain();
```

Qualifier Assessment Code Cave: x +

https://cognizant.tekstac.com/mod/vpl/forms/edit.php?id=110125&userid=137159#

File List Save Compile & Run Evaluate Reset Restore Description

File list

CodeCave

src

TrainerMain

TrainerMain.java

```
24 public List<String> findTrainerIdsWithLowPerformanceRating()
25 {
26     // Fill the code here
27
28     return null;
29 }
30
31
32
33
34 public static void main(String args[])
35 {
36     // 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
37
38     TrainerMain trainer=new TrainerMain();
39     List<String> list1=new ArrayList<String>();
40     Map<String, Float> map=new HashMap<String,Float>();
41     Scanner sc=new Scanner(System.in);
42     System.out.println("Enter number of records to be added:");
43     int n=sc.nextInt();
44     System.out.println("Enter the trainer records (Trainer id : Performance rating):");
45     String [] trainerDetails = new String[n];
46     for(int i=0;i<n;i++) {
47         trainerDetails[i] = sc.next();
48     }
49     for(int i=0;i<trainerDetails.length;i++) {
50         String[] a = trainerDetails[i].split(":");
51         map.put(a[0], Float.parseFloat(a[1]));
52     }
53     trainer.setTrainerMap(map);
54
55     System.out.println("Enter the trainer id to be searched");
56     String search=sc.next();
57     float rating=trainer.findPerformanceRatingOfGivenTrainerId(search);
58     if(rating==1)
59     {
60 }
```

Qualifier Assessment Code Cave: x +

https://cognizant.tekstac.com/mod/vpl/forms/edit.php?id=110125&userid=137159#

File List Save Compile & Run Evaluate Reset Restore Description

File list

CodeCave

src

TrainerMain

TrainerMain.java

```
44 String [] trainerDetails = new String[n];
45 for(int i=0;i<n;i++) {
46     trainerDetails[i] = sc.next();
47 }
48
49 for(int i=0;i<trainerDetails.length;i++) {
50     String[] a = trainerDetails[i].split(":");
51     map.put(a[0], Float.parseFloat(a[1]));
52 }
53 trainer.setTrainerMap(map);
54
55 System.out.println("Enter the trainer id to be searched");
56 String search=sc.next();
57 float rating=trainer.findPerformanceRatingOfGivenTrainerId(search);
58 if(rating==1)
59 {
60     System.out.println(search+" is an invalid trainer id");
61 }
62 else
63 {
64     System.out.println("Performance rating of trainer id "+search+" is "+rating);
65 }
66 list1=trainer.findTrainerIdsWithLowPerformanceRating();
67 if(list1.size()>=1)
68 {
69     System.out.println("Trainer id of the trainers with low rating are:");
70     for(String s:list1)
71     {
72         System.out.println(s);
73     }
74 }
75 else
76 {
77     System.out.println("No trainers have a low rating");
78 }
79
80 }
```