Real Champions Academy

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Based on: [Real Champions Academy](#)
Run: Yes **Evaluate**: Yes
Automatic grade: Yes

"Real Champions" is the city's well-known sports academy. They had held a 1000-meter running race, and many of the players had taken part in it. They wanted to find the time taken by the participant for a given participant name, as well as the participant who took the shortest time to complete the race. The event coordinator contacts a software developer to assist them in their process. As a software developer, you must create a Java program based on the requirements.

Component Specification: ParticipantInfo Class

Type (Class)	Attributes	Methods
ParticipantInfo	private Map<String, Integer> **participantMap**	Getter and setter methods for the attribute are included as the code skeleton.

Note: Here the participantMap, holds the Key as participantName and Value as timeTaken.

Requirement 1: Find the time taken for the given participant name.

Type (Class)	Methods	Responsibilities
ParticipantInfo	public int **findTheTimeTakenByTheParticipant(** String participantName)	This method accepts participantName as an argument. If the participantName is found in the Map, return the timeTaken by that participant. Else return -1. *Condition:* • *participantName is a case-insensitive*

Requirement 2: Filter the participantMap who finished the race in the shortest time.

Type (Class)	Methods	Responsibilities
ParticipantInfo	public List<String> **findTheParticipantWithTheShortestTime()**	This method filters the **participantMap** who took the shortest time to finish the race.

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.

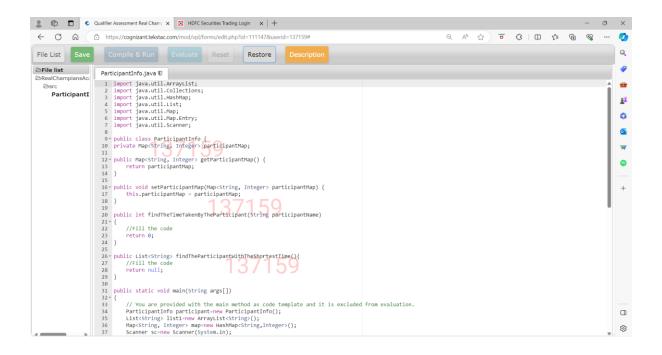**Sample Input / Output 1:**

Enter number of records to be added:

5

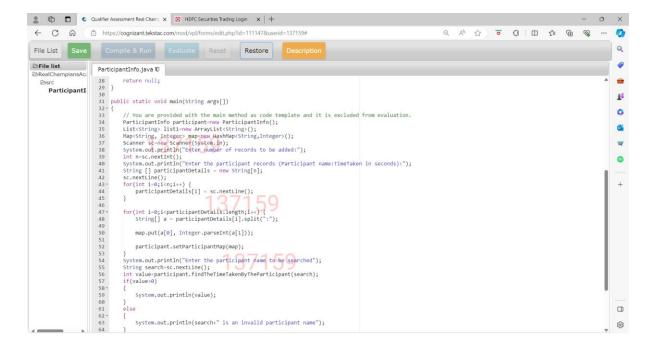Enter the participant records (Participant name:TimeTaken in seconds):

Miny:138

Teenu:126

Jack:145

Dave:120

Shery:156

Enter the participant name to be searched

Jack

145

Participant name with the shortest time taken:

Dave

Sample Input / Output 2:

Enter number of records to be added:

4

Enter the participant records (Participant name:TimeTaken in seconds):

Minu:138

Tiny:126

Stew:126

Dane:128

Enter the participant name to be searched

Jin

Jin is an invalid participant name

Participant name with the shortest time taken:

Stew

Tiny

File List   Save   Compile & Run   Evaluate   Reset   Restore   Description

File list
RealChampiansAc
  src
    ParticipantI

ParticipantInfo.java

```java
1  import java.util.ArrayList;
2  import java.util.Collections;
3  import java.util.HashMap;
4  import java.util.List;
5  import java.util.Map;
6  import java.util.Map.Entry;
7  import java.util.Scanner;
8
9  public class ParticipantInfo {
10 private Map<String, Integer> participantMap;
11
12 public Map<String, Integer> getParticipantMap() {
13     return participantMap;
14 }
15
16 public void setParticipantMap(Map<String, Integer> participantMap) {
17     this.participantMap = participantMap;
18 }
19
20 public int findTheTimeTakenByTheParticipant(String participantName)
21 {
22     //Fill the code
23     return 0;
24 }
25
26 public List<String> findTheParticipantWithTheShortestTime(){
27     //Fill the code
28     return null;
29 }
30
31 public static void main(String args[])
32 {
33     // You are provided with the main method as code template and it is excluded from evaluation.
34     ParticipantInfo participant=new ParticipantInfo();
35     List<String> list1=new ArrayList<String>();
36     Map<String, Integer> map=new HashMap<String,Integer>();
37     Scanner sc=new Scanner(System.in);
```

File List   Save   Compile & Run   Evaluate   Reset   Restore   Description

File list
RealChampiansAc
  src
    ParticipantI

ParticipantInfo.java

```java
28     return null;
29 }
30
31 public static void main(String args[])
32 {
33     // You are provided with the main method as code template and it is excluded from evaluation.
34     ParticipantInfo participant=new ParticipantInfo();
35     List<String> list1=new ArrayList<String>();
36     Map<String, Integer> map=new HashMap<String,Integer>();
37     Scanner sc=new Scanner(System.in);
38     System.out.println("Enter number of records to be added:");
39     int n=sc.nextInt();
40     System.out.println("Enter the participant records (Participant name:TimeTaken in seconds):");
41     String [] participantDetails = new String[n];
42     sc.nextLine();
43     for(int i=0;i<n;i++) {
44         participantDetails[i] = sc.nextLine();
45     }
46
47     for(int i=0;i<participantDetails.length;i++) {
48         String[] a = participantDetails[i].split(":");
49
50         map.put(a[0], Integer.parseInt(a[1]));
51
52         participant.setParticipantMap(map);
53     }
54     System.out.println("Enter the participant name to be searched");
55     String search=sc.nextLine();
56     int value=participant.findTheTimeTakenByTheParticipant(search);
57     if(value>0)
58     {
59         System.out.println(value);
60     }
61     else
62     {
63         System.out.println(search+" is an invalid participant name");
64     }
```

File List | Save | Compile & Run | Evaluate | Reset | Restore | Description

File list
RealChampiansAc:
  src
    ParticipantI

ParticipantInfo.java

```java
39        int n=sc.nextInt();
40        System.out.println("Enter the participant records (Participant name:TimeTaken in seconds):");
41        String [] participantDetails = new String[n];
42        sc.nextLine();
43        for(int i=0;i<n;i++) {
44            participantDetails[i] = sc.nextLine();
45        }
46
47        for(int i=0;i<participantDetails.length;i++) {
48            String[] a = participantDetails[i].split(":");
49
50            map.put(a[0], Integer.parseInt(a[1]));
51
52            participant.setParticipantMap(map);
53        }
54        System.out.println("Enter the participant name to be searched");
55        String search=sc.nextLine();
56        int value=participant.findTheTimeTakenByTheParticipant(search);
57        if(value>0)
58        {
59            System.out.println(value);
60        }
61        else
62        {
63            System.out.println(search+" is an invalid participant name");
64        }
65        list1=participant.findTheParticipantWithTheShortestTime();
66
67        System.out.println("Participant name with the shortest time taken:");
68        for(String s:list1)
69        {
70            System.out.println(s);
71        }
72 }
73
74 }
75
```