Hyatt IQ Level

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Based on: Hyatt IQ Level
Run: Yes **Evaluate**: Yes
Automatic grade: Yes

In the city, **"Hyatt"** is a well-known teaching platform organization. They held the event to test the participants' IQ levels, and many of them took part. They now want to find the participant's IQ level based on the participant's name and filter the participant with the highest IQ level. The manager contacts a software developer to assist them with 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 in skeleton.

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

Requirement 1: Find the iQLevel of the participant.

Type (Class)	Methods	Responsibilities
ParticipantInfo	public int **findTheIQLevelforGivenParticipantName**(String participantName)	This method accepts parti... argument. If the particip... the Map, return their iQL... *Condition*: *participantName is a cas...*

Requirement 2: Find the participants with the highest IQ level.

Type (Class)	Methods	Responsibilities
ParticipantInfo	public List<String> **findTheHighestIQLevel**()	This method finds the part... highest IQ level, adds it to... the list.

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:

6

Enter the participant records (Participant name:IQ level):

Wade:89

Dave:90

Seth:40

Ivan:74

Riley:80

Gilbert:68

Enter the participant name to be searched

Ivan

74

Participant name with the highest IQ level are:

Dave

Sample Input/Output 2:

Enter number of records to be added:

6

Enter the participant records (participant name:IQ level):

Wade:89

Dave:90

Seth:40

Ivan:74

Riley:90

Gilbert:68

Enter the participant name to be searched
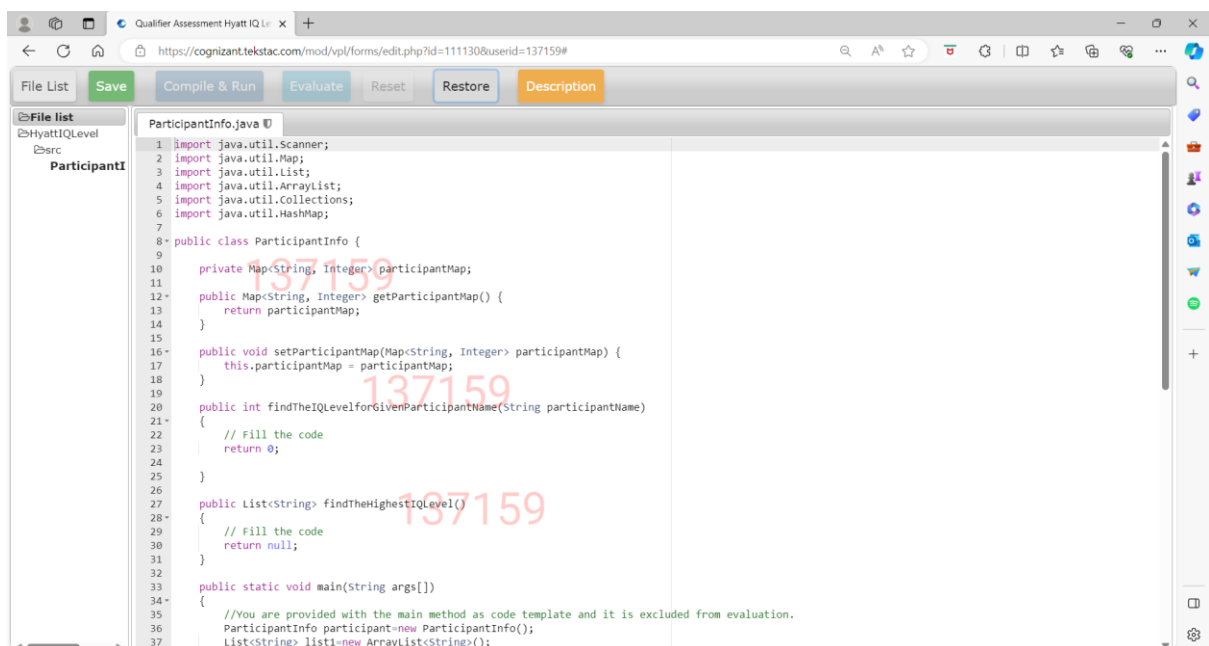
John

John is an invalid participant name

Participant name with the highest IQ level are:

Dave

Riley

File List   Save   Compile & Run   Evaluate   Reset   Restore   Description

File list
HyattIQLevel
  src
    ParticipantI

ParticipantInfo.java

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