ZMovies

**Grade settings**: Maximum grade: 100  
**Disable external file upload, paste and drop external content**: Yes  
**Based on**: [ZMovies](https://cognizant.tekstac.com/mod/vpl/view.php?id=91142)  
**Run**: Yes **Evaluate**: Yes  
**Automatic grade**: Yes

ZMovies is a popular website for reviewing movies. They have planned to provide movie details and ratings through an online platform so they can encourage the youths to watch movies in theatres instead of on an OTT platform. The company hires a software developer to help with their process. You, being the software developer, develop a Java program based on the requirement.

**Component Specification: MovieMain Class**

|  |  |  |
| --- | --- | --- |
| **Type (Class)** | **Attributes** | **Methods** |
| **MovieMain** | Map<String, Float> movieDetailsMap | Getter and setter methods for the attribute are included in the code skeleton. |

***Note:****Here the movieDetailsMap, holds the Key as movieName and Value as rating.*

**Requirement 1: Filter the movie record based on the movie name**

|  |  |  |
| --- | --- | --- |
| **Type (Class)** | **Methods** | **Responsibilities** |
| **MovieMain** | public float **findMovieRating**(String movieName) | This method accepts movie name as argument. If the movie name is found in the Map, return the movie's rating. Else return -1.  ***condition:****movieName is case insensitive* |

**Requirement 2: Filter the movies based on the highest rating**

|  |  |  |
| --- | --- | --- |
| **Type (Class)** | **Methods** | **Responsibilities** |
| **MovieMain** | public List<String> **findMoviesWithHighestRating()** | This method filters the movies and returns the list of movie names that satisfy below-mentioned condition  **Condition :** All the movie whose rating is greater than 4 is considered as the Highest rating and it needs to be added 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:

**3**

Enter the Movie records (Movie Name : Rating):

**Spider Man No Way Home:5.0**

**The Dark Knight Rises:3.0**

**Inception:2.0**

Enter the movie name needs to be searched

**Inception**

2.0

The names of the movies with the highest rating are:

Spider Man No Way Home

**Sample Input/Output 2:**

Enter number of records to be added:

**2**

Enter the Movie records (Movie Name : Rating):

**Interstellar:3.0**

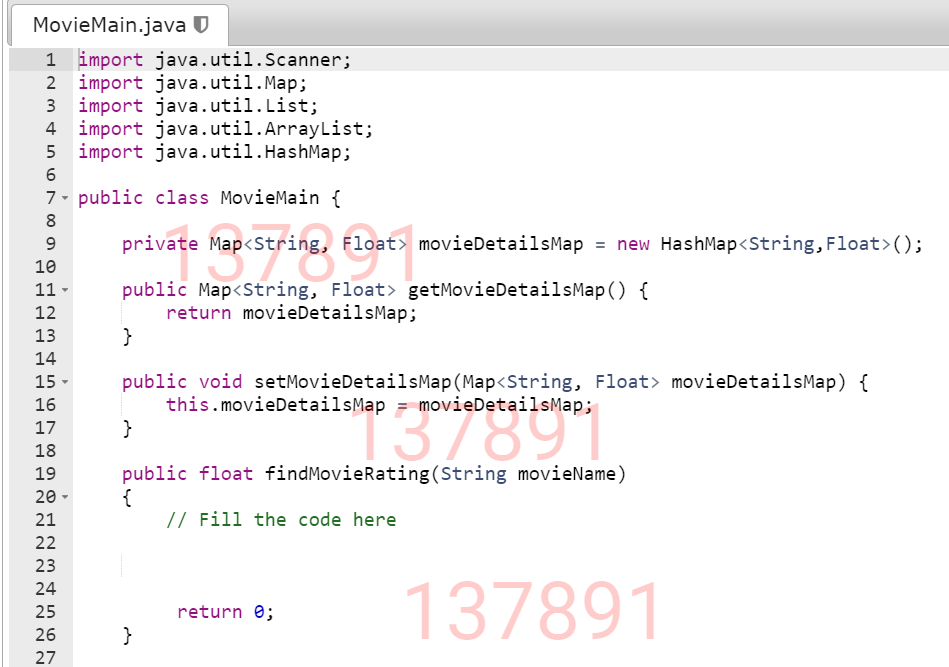
**Man of Steel:4.0**

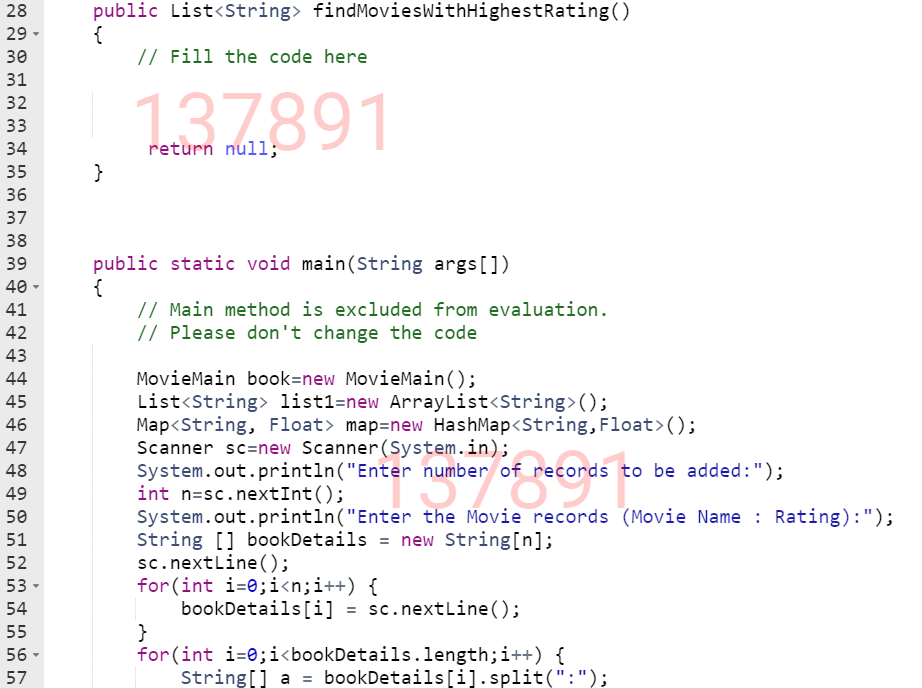
Enter the movie name needs to be searched

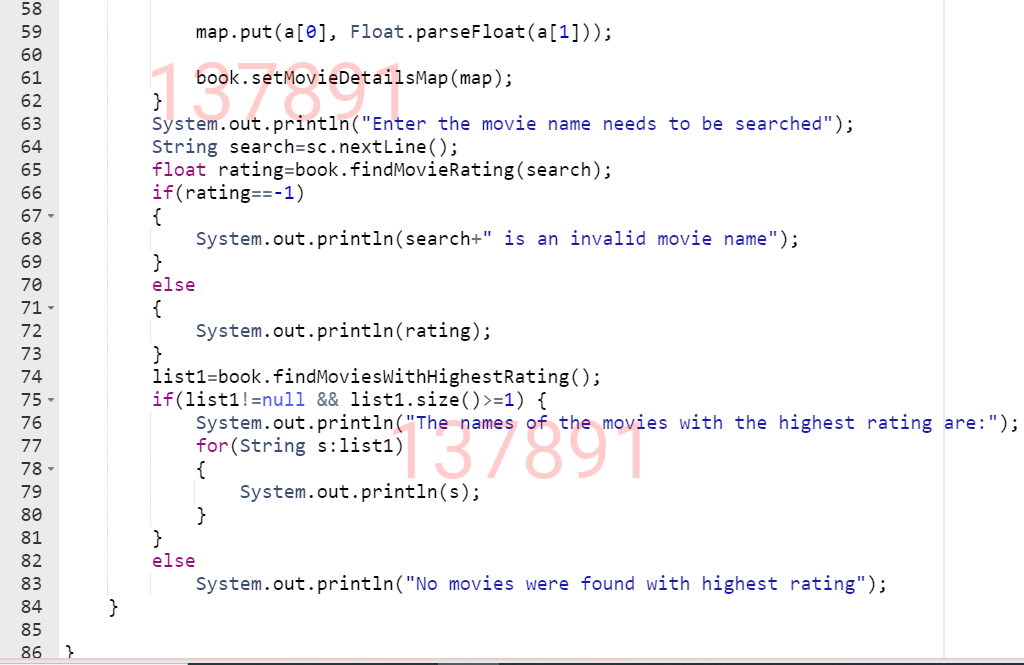
**Tenent**

Tenent is an invalid movie name

No movies were found with highest rating







Hyatt IQ Level

**Grade settings**: Maximum grade: 100  
**Disable external file upload, paste and drop external content**: Yes  
**Based on**: [Hyatt IQ Level](https://cognizant.tekstac.com/mod/vpl/view.php?id=91139)  
**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 included in the code 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 participantName as an argument. If the participantName is found in the Map, return their iQLevel. Else return -1.  ***Condition****:*  ***participantName****is a case insensitive.* |

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

|  |  |  |
| --- | --- | --- |
| **Type (Class)** | **Methods** | **Responsibilities** |
| **ParticipantInfo** | public List<String> **findTheHighestIQLevel()** | This method finds the participants with the highest IQ level, adds it to the list and returns 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

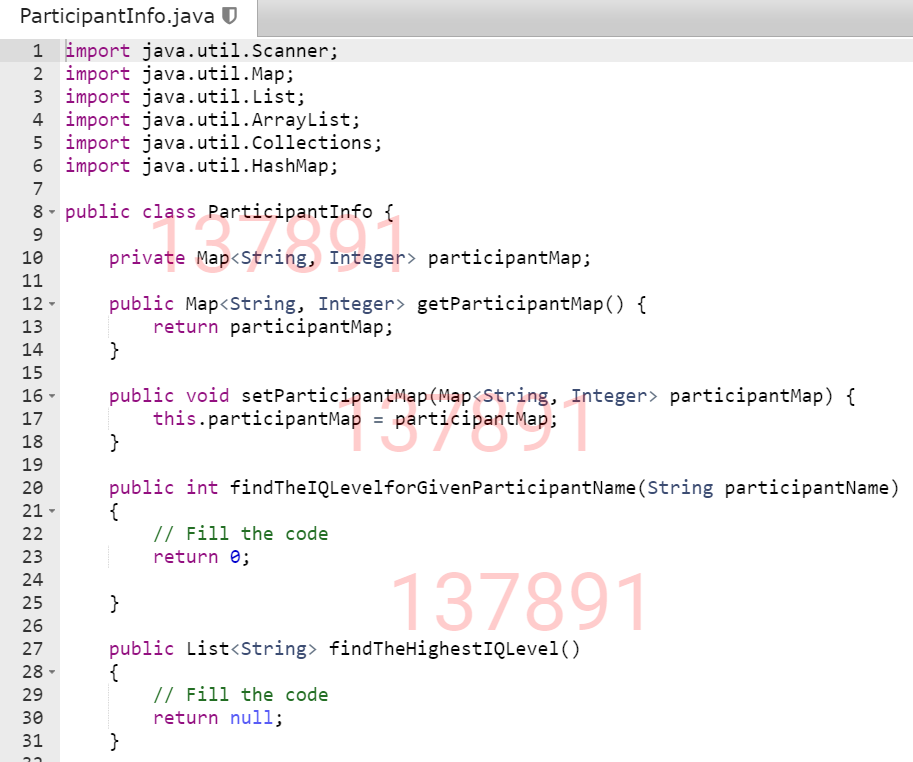
**John**

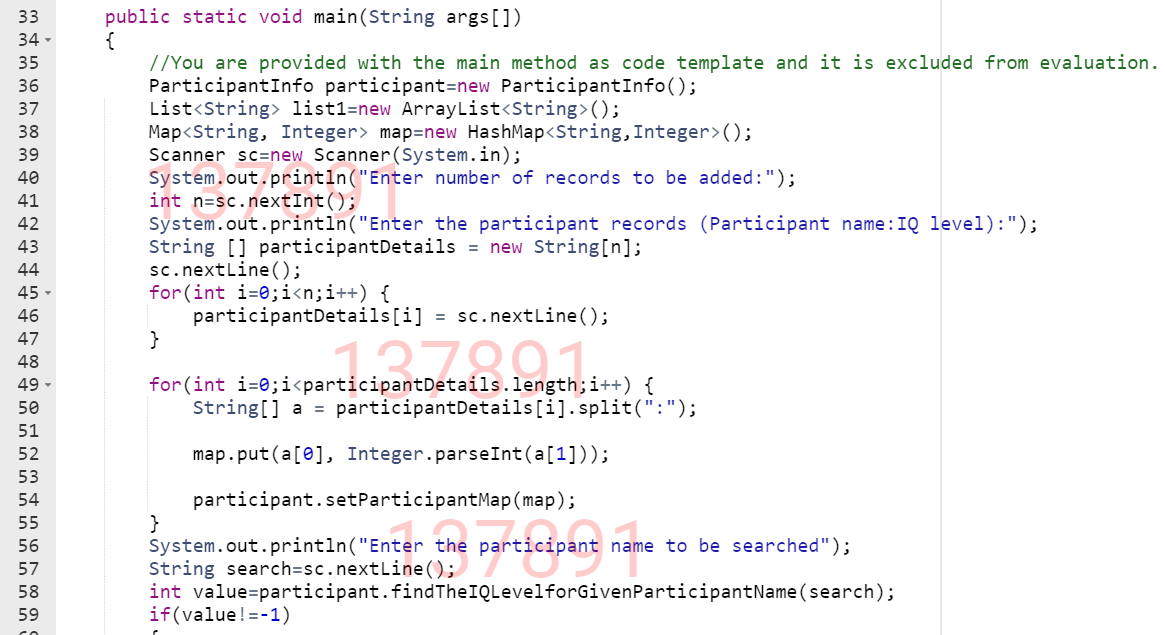
John is an invalid participant name

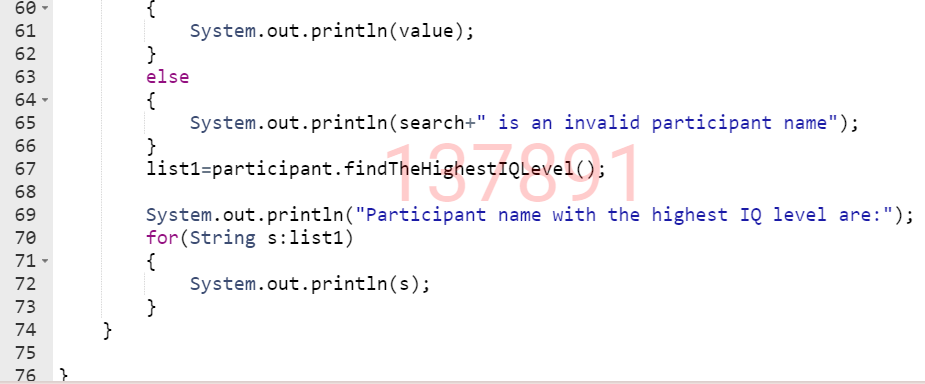
Participant name with the highest IQ level are:

Dave

Riley







Real Champions Academy

**Grade settings**: Maximum grade: 100  
**Disable external file upload, paste and drop external content**: Yes  
**Based on**: [Real Champions Academy](https://cognizant.tekstac.com/mod/vpl/view.php?id=91138)  
**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

