



UNIVERSITY OF KWAZULU-NATAL

COMP102: Compute Programming
Practical Test

Thursday, 3 November 2022

Examiner: Sibonelo Dlamini
Moderator: Rosanne Els
Time Allocated: 180 minutes
Total Marks: 75

Instructions

- Answer **both** questions
- Write your **student number** as a comment on the top of all **.java** files
- You may **NOT** copy code from a classmate, or give your code to classmate to copy
- You may **NOT** steal code from the Internet

Scenario

UKZN hosted a mini-track day, where 208 students participated in 5 sprints over 100 meters. These students were taken from all five UKZN campuses, namely: Howard, Westville, PMB, Edgewood and Medical School. You have been given a data text file called **athletes.txt** which contains the results of the five races.

The first line of the text file contains the number of students who participated in the track day. Each subsequent line contains the following pieces of information, separated by a semi-colon: **name; surname; campus; sprint time 1; sprint time 2; sprint time 3; sprint time 4; sprint time 5**.

An excerpt of the file is shown below:

```
208
Kriven;Govender;Westville;19.34;20.99;19.25;20.16;20.34
Nicholas;Culverwell;Howard;17.83;15.1;15.66;15.89;17.52
Yusuf;Seedat;Howard;17.49;14.06;16.9;16.1;16.91
Rakeen;Ramdeen;PMB;16.29;14.66;17.99;14.23;14.34
Daniel;Hogg;Edgewood;10.39;12.35;12.56;10.45;11.17
```

QUESTION 1: Athlete.java [15 marks]

- 1.1 A method called `getAverageTime()` has been created in **Athlete.java**. Complete this method so that it calculates the average sprint time that a student athlete achieved over their 5 sprints. The method should return this time. (5)
- 1.2 A method called `getBestTime()` has been created in **Athlete.java**. Complete this method so that it determines the best time a student athlete achieved among the 5 sprints they did. The method should return this value. (5)
- 1.3 A `toString()` method has been created in **Athlete.java**. Complete this method so that it returns a string which shows the state of a student object. The string returned should display the details of the object as follows: (5)

```
Name: Asemahle  
Surname: Mpofana  
Campus: Howard  
Times: [12.24] [12.6] [14.65] [14.65] [13.56]
```

[15]

QUESTION 2: PerformanceAnalyser.java [60 marks]

- 2.1 Within the `main()` method of **PerformanceAnalyser.java**, create an array of Athlete objects which correspond to the student athlete information in the `athletes.txt` file. A single Athlete object should be created for each line in the text file, and then stored in this array. (15)
- 2.2 A method called `getFastestStudent()` has been created in **PerformanceAnalyser.java**. Complete this method so that it displays the student athlete who recorded the fastest sprint, i.e. the shortest best time. The method should display the name, surname, and best time of this student, as formatted below: (10)

```
Alissa Chetty is the fastest student. Their best  
sprint time is: 10.0
```

- 2.3 A method called `showGoodHowardSprinters()` has been created in **PerformanceAnalyser.java**. Complete this method so that it firstly calculates the average sprint time achieved by all athlete who participated in the track day. Then it should display all athletes from **Howard** campus whose average sprint time is below this overall average. (15)

An incomplete sample output is shown below:

```
Name: Thandokazi  
Surname: Phunzana  
Campus: Howard  
Times: [13.93] [14.31] [14.95] [12.41] [12.83]
```

```
Name: Nonkanyiso  
Surname: Ngwezi  
Campus: Howard  
Times: [10.18] [12.19] [12.36] [10.02] [12.07]
```

```
Name: Zibusiso  
Surname: Mnguni  
Campus: Howard  
Times: [11.38] [12.27] [11.52] [11.78] [11.35]  
....
```

- 2.4 A method called `searchStudents()` has been created in **PerformanceAnalyser.java**. Complete this method so that it firstly calculates the average sprint time of those students who **name** begins with the character supplied as the second parameter to the method. Of this group, it should then display the **subset** of student athletes whose average sprint time is **less than** the group's average. (20)

Given the letter "A" as the second parameter, the following **incomplete** output should be displayed:

```
Average time for student names beginning with A:  
15.15474074074074  
Name: Alissa  
Surname: Chetty  
Campus: Medical School  
Times: [11.1] [10.55] [11.33] [10.0] [10.84]
```

```
Name: Andile  
Surname: Mncube  
Campus: Edgewood  
Times: [14.97] [14.14] [14.4] [14.52] [13.42]
```

```
Name: Asemahle  
Surname: Mpofana  
Campus: Howard  
Times: [12.24] [12.6] [14.65] [14.65] [13.56]  
...
```

Given the letter "C" as the second parameter, the following **incomplete** output should be displayed:

```
Average time for student names beginning with C:  
14.464
```

Name: Caylan
Surname: Sardhavu
Campus: Medical School
Times: [12.57] [14.09] [12.54] [14.52] [13.25]

Name: Casey
Surname: Pillay
Campus: Howard
Times: [12.61] [11.0] [10.77] [11.69] [12.76]

Name: Chirag
Surname: Kassie
Campus: Westville
Times: [10.23] [11.93] [12.63] [11.78] [12.91]

[60]