LEARNING PROFILE FOR ASSIGNMENT#1, QUESTION#4

BanffMarathonRunner.java

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Name* | *:* | *Tyler Lucas* | *Due Date* | *:* | *N/A* |
| *Student ID* | *:* | *3305203* | *Submission Date* | *:* | *2017/05/09* |

# 1. Problem Statement

A group of AU friends decide to run the Banff, Alberta, Marathon. Their names, times (marathon completion time in minutes), and number of years participated are given below:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Elena Brandon | 341 | 1 |
| 2 | Thomas Molson | 273 | 2 |
| 3 | Hamilton Winn | 278 | 5 |
| 4 | Suzie Sarandin | 329 | 7 |
| 5 | Philip Winne | 445 | 9 |
| 6 | Alex Trebok | 275 | 3 |
| 7 | Emma Pivoto | 275 | 4 |
| 8 | John Lenthen | 243 | 1 |
| 9 | James Lean | 334 | 1 |
| 10 | Jane Ostin | 412 | 1 |
| 11 | Emily Car | 393 | 4 |
| 12 | Daniel Hamshire | 299 | 4 |
| 13 | Neda Bazdar | 343 | 3 |
| 14 | Aaron Smith | 317 | 6 |
| 15 | Kate Hen | 265 | 8 |

Extend the AddressBook class from Problem 1 to store the additional data. Now, write a method to find the fastest runner. Print the name, address, and his/her time (in minutes) on three separate lines.

Find the second fastest runner. Print the name, address, his/her time (in minutes), and the difference in time with the fastest runner.

Compute the average time of completion taken by these runners.

Finally, print the name and number of years participated for each runner if the runner’s time of completion is equal to or better than the average time of completion.

# 2. Description of the Code

Outputs temperature in Fahrenheit given an input temperature in Celsius.

# **3.** Errors and Warnings

Table 1: List of Errors and Warnings Encountered in the Program

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Errors / Warnings** | **Details** | **How I solved them** |
| 1 | CelsiusToFahrenheit class wasn’t found in CelsiusToFahrenheit project. | [v. 1.0] I had set the main class as "private". | I changed the class and main method from “private” to “public”. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 4. Sample Input and Output

[Version 1.2-1.3, input “0”]

Enter temperature in integer degrees Celsius: 0

0 degrees Celsius is equivalent to 32 degrees Fahrenheit.

[Version 1.2-1.3, input “60.5” (float type input)]

Enter temperature in integer degrees Celsius: 60.5

Exception in thread "main" java.util.InputMismatchException

at java.util.Scanner.throwFor(Scanner.java:864)

at java.util.Scanner.next(Scanner.java:1485)

at java.util.Scanner.nextInt(Scanner.java:2117)

at java.util.Scanner.nextInt(Scanner.java:2076)

at CelsiusToFahrenheit.main(CelsiusToFahrenheit.java:34)

C:\Users\tyblu\Documents\repos\comp268-random\CelsiusToFahrenheit\nbproject\build-impl.xml:1040: The following error occurred while executing this line:

C:\Users\tyblu\Documents\repos\comp268-random\CelsiusToFahrenheit\nbproject\build-impl.xml:805: Java returned: 1

# 5. Discussion

The first error, where a class couldn’t be found in the project, was caused by setting either or both the class and the main method to private. I first ran into this error when attempting the *HelloWorld* sample program[[1]](#footnote-1). Having read the class Style Guide in which it says “Create private fields with getters/setters rather than leaving fields public,” as well as Controlling Access to Members of a Class[[2]](#footnote-2) in which it says “Use private unless you have a good reason not to,” I mistakenly thought this applied to the main class and method as well, as I’m still not sure what the differences are between a class, method, and object. Searching online didn’t reveal a solution right away, as few experienced programmers would think that something this simple could go awry, but I eventually found the answer here: <https://goo.gl/P2OdMJ>. Of course, the next page in the textbook had the answer as well:

The word “public” in the first line of main() means that this routine can be called from outside the program. This is essential because the main() routine is called by the Java interpreter, which is something external to the program itself.[[3]](#footnote-3)

I’ll have to keep an eye out to

1. (Eck, 2014, p. 21) [↑](#footnote-ref-1)
2. (Oracle, 2015) [↑](#footnote-ref-2)
3. (Eck, 2014, p. 22) [↑](#footnote-ref-3)