|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **LEARNING PROFILE FOR ASSIGNMENT#\_1\_\_\_\_\_ AND QUESTION#\_4\_\_\_\_\_\_\_** | | | | | |
| *Name* | *:* | *Steven Morrissey* | *Due Date* | *:* |  |
| *Student ID* | *:* | *3300222* | *Submission Date* | *:* |  |

**1. Problem Statement:**

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:**

Used a static initialization for the list of runners, as for the purposes of this exercise the information will not be changing. For the getFastest and getSecondFastest methods, I used the efficiency of Java 8 method references combined with Arrays utility to sort the arrays and take the [0] and [1] indexes respectively for each method.

**3. Errors and Warnings:**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Errors / Warnings** | **Details** | **How I solved them** |
| 1 | none |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

**4. Sample Input and Output:**

**Case 1: Given an Array of BanffMarathonRunner, the name, address and time are printed out on separate lines for fastest runner; name, address, time and difference with fastest is printed out for second fastest runner; average time of all runners is printed out; name and # of years is printed out for all runners faster than the average time.**

**5. Discussion:**

None, straight forward