Module: Object Oriented Development

CA: CA1 - Project

Value:

Due Date: See Moodle

Objectives:

- To design, implement and test an Activity Tracker application
- Demonstrate the use of comparable, the comparator interface, anonymous inner classes and lambda expressions
- Incorporate collection classes where relevant
- Demonstrate the ongoing use of version control
- Apply DRY program design principles (DRY=Don't Repeat Yourself)
- Demonstrate the use of enums

Requirements

You are required to develop an activity tracker application to allow an end user to view and analyse their activity in a variety of ways. Your system will allow the user to view some or all of their activities sorted in a number of ways (outlined in the Function Requirements section). It should also allow the user to view statistics based on their performance to date. Data will be entered into the system through a CSV file including the following fields:

- Type of Activity
- Duration (in minutes)
- Date
- Distance (in kilometers)
- Average Heart Rate

In addition to these fields your application will be expected to also calculate a number of other values including:

- Energy expended based on the average kilometres per hour
- Calories burned based on energy expended, type of activity and duration

The overall system should demonstrate a thorough understanding of all of the concepts that have been discussed in class this semester.

Functional Requirements.

- Allow the user to upload a set of activities from a CSV file. *(Duplicates should not be stored)*
- Allow the user to view their activity in the following order:
    - Calories burned (Descending)
    - Date (Ascending/Descending)

    - Activity Duration (Ascending/Descending)
    - Type of Activity.
    - Distance (Ascending/Descending)

- Allow the user view a subset of their activity based on specific fields
    - Activity type
    - Above a minimum distance
    - Type of energy expended
    - Above a minimum duration

- Find specific activity based on their *Natural ordering* using the binary search function of the Collections class

- View statistics on their overall performance
    - Average distance per activity
    - Average calories burned

CSV File

Data will be stored in a CSV file that must be read in to your application. The data will be presented in the following format

Activity Type, Date, Duration, Distance, Average heart rate

- *Activity Type*: One of the known activity types {Running, Swimming, Cycling}
- *Date*: Given in the following format **dd/mm/yyyy**
- *Duration:* Time taken to perform the activity in minutes
- *Distance:* Distance Travelled in kilometers?
- *Average Heart Rate:* The average heart rate during the activity

Additional Fields Formulae

In addition to the fields read in from the CSV file, your system must also calculate the following:

- *Intensity:* Intensity of the activity performed. This can be one of the following values {Very Light, Light, Moderate, Vigorous, Very Vigorous} *See appendix A for Intensity based on activity type.*

- *Calories Burned:* The number of calories burned during the activity. This can be calculated as follows

    - (Intensity Value * duration in minutes)
- *The intensity values can be found in appendix B*

Notes:

- You are required to use a combination of natural ordering, separate Comparator class(es), an anonymous inner class, and a number of lambdas

Appendix A: -

Kilometres Per hour	**Swimming**
0.5	Very Light
1.25	Light
2	Moderate
2.75	Vigorous
3.5	Very Vigorous

Kilometres Per hour	**Running**
> 4 (2.5)	Very Light
4-8	Light
8-12 (6)	Moderate
12-16	Vigorous
16-24 (12.5)	Very Vigorous

Kilometres Per hour	**Cycling**
> 8	Very Light
8 - 16	Light
17 - 25	Moderate
25 – 33	Vigorous
33 – 40	Very Vigorous

Appendix B: -

Intensity	**Swimming**	**Running**	**Cycling**
Very Light	**5**	**4.1**	**2**
Light	**6.3**	**7.2**	**5**
Moderate	**7.6**	**10**	**7**
Vigorous	**8.9**	**15.4**	**13**
Very Vigorous	**10.2**	**20.8**	**15**