Homework 2 : Small Programs

(Deadline as per Canvas)

HW deadline as per Canvas.

This homework deals with the following topics:

- Variables and data types

The Assignment

For this Java assignment, there are the five mini-programs we want you to code. Each mini-program will be in its own class, and all of the code you write should be in the *main* method of each class file.

We are providing you with the five class files (in MiniPrograms.zip file). Download the provided class files and put them in the "src" folder in your Java project. Follow the "Importing a Java Program into Eclipse.pdf" document for how to import a Java Program into a Java project in Eclipse.

The *main* method in each class has been defined for you, but without most of the code. See the comments for instructions on what each program is supposed to do and how to write the code. It should be clear enough. In some cases, we have provided hints to help you get started.

For example, we have described a "Add Five Numbers" program for you (see below) which prints the sum (as a double) of five given numbers, one by one. Read the comments, which explain what the code is supposed to do. Then write your code where it says "// TODO" to implement the program and replace the placeholder. You'll do this for each program defined in each class file.

These are the five programs that need to be implemented, each one in the *main* method of its own class.

1. *Add Five Numbers*
    a. Write your code in the provided *AddFiveNumbers.java* class file
    b. Print the total sum (as a double) of five given numbers, one by one
    c. Output from the program should look something like this:

Sum: 3.2
Sum: 4.2
Sum: 3.2
Sum: 3244.2
Sum: 3234.2999999999997

d. Note, (for now) do not be concerned with rounding double values. (We'll learn how to do this later.)

2. *Calculate Health Data*
   a. Write your code in the provided *CalculateHealthData.java* class file
   b. Based on a given age in years, calculate the number of days old, number of minutes old, and the number of times your heart has beat.
   c. Output from the program should look something like this:

      You are 7670 days old. You are 11044800 minutes old.
      Your heart has beat 795225600 times.

   d. Note, your program should consider leap years when calculating the number of days old.

3. *Convert Seconds to Time*
   a. Write your code in the provided *SecondsToTime.java* class file
   b. Convert a given number of seconds to hours:minutes:seconds.
   c. For example, if the given number of seconds is 1432, print output in the format: 0:23:52
   d. If the given number of seconds is 0, print output in the format: 0:0:0

4. *Compute Total Cost*
   a. Write your code in the provided *ComputingTotalCost.java* class file
   b. Based on given prices, compute the total cost of a particular number of drinks and tacos.
   c. For example, if a drink costs 2.10 dollars and a taco costs 3.43 dollars, the total cost of 4 drinks and 6 tacos is 28.98.

5. *Known People*
   a. Write your code in the provided *KnownPeople.java* class file
   b. Based on a given number of your friends, calculate the total number of known people.

c. For example, if you have 4 friends, then each of them knows 4 people. Then each of those people also knows 4 people. This means there are 1 (yourself) + 4 + 16 + 64 = 85 total known people.
d. Output from the program should look something like this:

```
You have 4 friends.
Each of them knows 4 people, which means 16 more people.
Then each of those people also knows 4 people, which means 64 more people.
This means, there are a total of 85 known people (including yourself).
```

Write comments using // for any non-trivial lines of code.

Submission

To complete the assignment, write the mini-programs as described in each class file. Submit all five completed class files and submit as a single .zip file.

Evaluation

Points:

1. Did you set up the files correctly? Do they compile and is everything named correctly? (2,5 pts)
2. Does your code function? Does it do what the specifications require? (18 pts)
    a. AddFiveNumbers - 5pts
    b. CalculateHealthData - 3pts
    c. ComputingTotalCost - 3pts
    d. KnownPeople - 3pts
    e. SecondsToTime - 4 pts
3. Did you write good comments (including javadoc comments for your classes) throughout your code? (2.5 pts)
4. Did you follow good style conventions? This includes camelCase for variables and proper indentation. (2.5 pts)
5. Did you submit the correct files “AddFiveNumbers.java”, “CalculateHealthData.java”, “ComputingTotalCost.java”, “KnownPeople.java”, and “SecondsToTime.java” as a single zip file? (2.5 pts)