The City College of New York **Department of Compute Science**
CSc 221: Software Design Laboratory

Assignment 1 – Spring 2021

Due Date: *by Sunday February 28, 2021 11:59PM*
How to submit: *upload JAVA files to Blackboard*

Please note:

- ✓ This is an individual assignment; please do your own work. Sharing and/or copying code in part or as a whole with/from others will result in a grade of 0 and disciplinary actions for all involved parties. If you run into problems and have done your best to solve them, please talk to me during office hours or by e-mail.
- ✓ There is a 25% grade deduction for every day the assignment is late unless prior permission is granted.

Preamble

Body mass index (BMI) is a measure of the body's fat based on height and weight. BMI applies to adult men and women and considered a convenient rule of thumb to broadly categorize a person as underweight, normal weight, overweight, or obese based on tissue mass (muscle, fat, and bone) and height. That categorization is the subject of some debate about where on the BMI scale the dividing lines between categories should be placed. Commonly accepted BMI ranges are underweight (under 18.5 kg/m2), normal weight (18.5 to 25), overweight (25 to 30), and obese (over 30)[I]. For more information, please visit https://en.wikipedia.org/wiki/Body_mass_index

In this assignment, you will develop a Java command-line program to compute a person's BMI based on the criteria described earlier.

Requirements:

1. Your program must compile, run, and conform to the Java conventional programming rules
2. Comment your code:
   - ✓ Place Javadoc comments for the class' definition and for each of the class' members (variables and methods)
   - ✓ Place non-Javadoc comments for major steps in your code
3. Submit a <u>separate</u> source file for each of the following two classes:
   1. $PersonInfo$ – a container for the data and operations relating to a person's weight.
   2. $TestPersonInfo$ – a class which contains the $main$ method and performs unit testing on the previous class.
3. You must use $String\ formatters\ (i.e.printf(\dots)\ and\ String.format(\dots))$ for printing. Your solution should NOT contain the methods $print(\ )$ OR $println(\ )$.
4. Class $PersonInfo$:
   a. Four variables for $Name$ $(String)$, $year\ of\ birth$ (Int), $height$ $(Double)$, and $weight$ $(Double)$.
   b. Default constructor to initialize the four variables to their default values ("" for strings and 0 for numeric).
   c. Non-default constructor to initialize the four variables using 4 constructor parameters.
   d. Accessors (setters) and mutators (getters) methods for all four variables – 8 methods in total.
   e. $calculateAge(\ )$ to return the person's age.
      ⇨ *In order to compute the age, you will need to <u>programmatically</u> obtain the current year. <u>DO NOT</u> hard-code the current year as part of you class. Use the following Java statement to retrieve the current year:*
      $$new\ GregorianCalendar(\ ).get(GregorianCalendar.YEAR)$$
   f. $calculateBMI(\ )$ to return the person's BMI:
      ⇨ $bmi = weight/height^2$
   5. Class $TestBMI$:
      Remember:

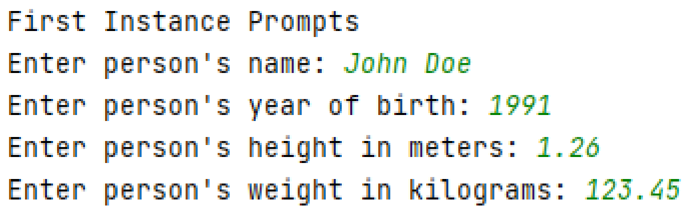
[I] https://en.wikipedia.org/wiki/Body_mass_index

  - ✓ Show a message string before the prompt.
  - ✓ Using $printf(\quad)$ ONLY
- ✓ Create a method called $classifyBMI()$; the method returns one of the following strings:
  - ✓ "Underweight": if BMI is under 18.5
  - ✓ "Normal Weight": if BMI is between 18.5 and 25
  - ✓ "Overweight": if BMI is between 25 and 30
  - ✓ "Obese": if BMI is more than 30
- ✓ Create an instance of class $PersonInfo$ using the $\underline{default}$ constructor. Use the mutator methods to assign values to the class' 4 private members. The test values are shown in Figure 1.
- ✓ Prompt the user to enter the person's $full\ name, year, height, and\ weight$. These values will be used in the next step. Sample test values are shown in Figure 2.
- ✓ Create an instance of $PersonInfo$ using the <u>non-default</u> constructor. Use the values from the previous step.
- ✓ Using $printf(\quad)$ print the information such that it is formatted as shown in the example in Figure 1 and Figure 2. All leading labels should have column widths of 16 characters.

Grading:

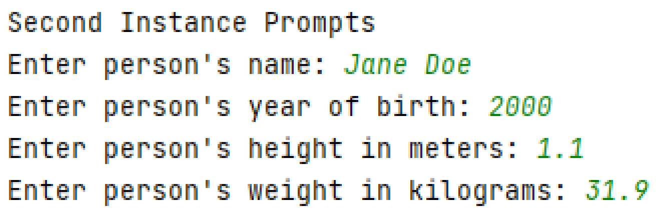
Item	Points
Comments (Javadoc and major steps)	10
$PersonInfo$ class (Compiles and runs)	
4 variables	2
Accessor and mutator methods	8
Constructors	10
$calculateAge(\quad)$	10
$calculateBMI(\quad)$	10
$TestBMI$ class (Compiles and runs)	
Instance of $PersonInfo$ using the default constructor	3
Prompt and capture of input	4
Setters	4
Instance of $PersonInfo$ using the non-default constructor	10
Prompt and capture of input	4
Printing person's info to command prompt from both instances	15
Correct output	10
	50

Figures:



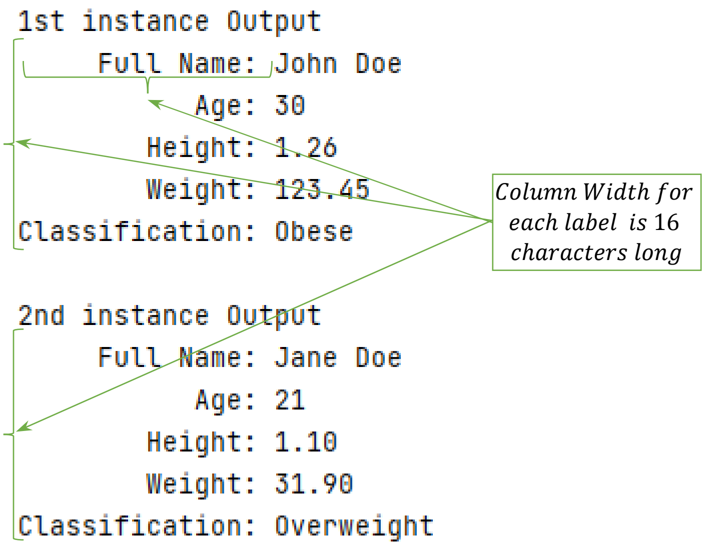
```
First Instance Prompts
Enter person's name: John Doe
Enter person's year of birth: 1991
Enter person's height in meters: 1.26
Enter person's weight in kilograms: 123.45
```

Figure 1: Sample Run 1



```
Second Instance Prompts
Enter person's name: Jane Doe
Enter person's year of birth: 2000
Enter person's height in meters: 1.1
Enter person's weight in kilograms: 31.9
```

Figure 2: Sample Run 2



```
1st instance Output
     Full Name: John Doe
           Age: 30
        Height: 1.26
        Weight: 123.45
Classification: Obese

2nd instance Output
     Full Name: Jane Doe
           Age: 21
        Height: 1.10
        Weight: 31.90
Classification: Overweight
```

Column Width for each label is 16 characters long

Figure 3: output of program