Assignment 1 – Spring 2020

Body mass index (BMI) is a measure of body fat based on height and weight that applies to adult men and women. The BMI is a convenient rule of thumb used 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). For more information please visit https://en.wikipedia.org/wiki/Body_mass_index

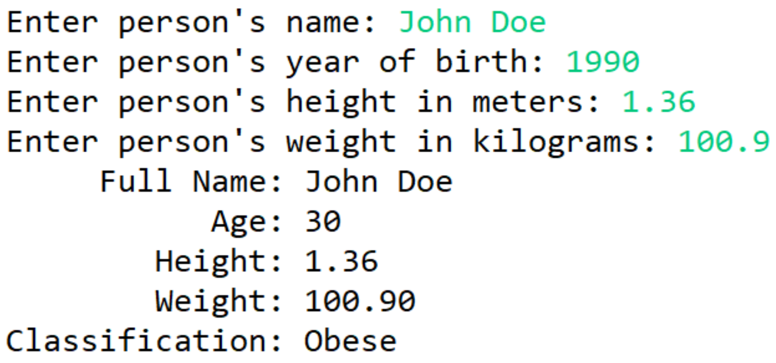
Assignment's Requirements:

1. Your program should compile and run in order to be graded
2. Submit two separate classes as described below:
   a. $PersonWeight$ – a container and operations class for weight information.
   b. $TestPersonWeight$. This class contains the main method to perform unit testing on the previous class.
3. You must use $String\ formatters\ \left(i.e.\,printf(\ldots)\right)$ for ANY output printing. Your solution should NOT contain the methods $print(\quad)$ OR $println(\quad)$.
4. Class $PersonWeight$:
   a. Four variables for Full Name (String), year of birth (Int), height (Double), and weight (Double).
   b. Default constructor to initialize the four variables to some default values – you may pick any values here.
   c. Non-default constructor to initialize the four variables using the constructor's parameters.
   d. Accessors (setters) and mutators (getters) methods for all four variables – 8 methods in total.
   e. $computeAge(\quad)$ to return the person's age.
      ⇨ *In order to compute the age, you will need to <u>programmatically</u> obtain the current year. <u>DONOT</u> hard-code the current year as part of you class. Use the following Java statement to retrieve the current year:*
      $$new\ GregorianCalendar(\quad).get(GregorianCalendar.YEAR)$$
   f. $computeBMI(\quad)$ to return the person's BMI:
      ⇨ $bmi = weight/height^2$
5. Class $TestPersonWeight$:
   a. Remember:
      ✓ Show a message string before the prompt.
      ✓ Using $printf(\quad)$ ONLY
   b. 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
   c. Create an instance of class $PersonWeight$ using the $\underline{default}$ constructor. **Use the mutator methods to assign values to the class' 4 private members.**
   d. Prompt the user to enter the person's full name, year, height, and weight. **These values will be used in the next step.**
   e. Create an instance of $PersonWeight$ using the <u>non-default</u> constructor. **Use the values from the previous step.**
   f. 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:

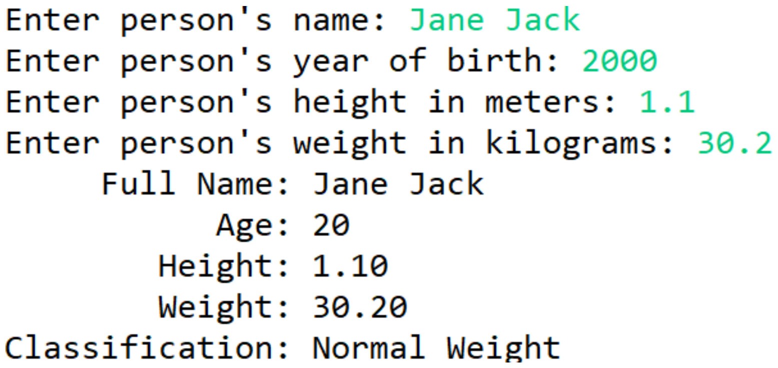
Item	Points
$PersonWeight$ class (Compiles and runs)	
4 variables	5
Accessor and mutator methods	5
Constructors	20
$computeAge(\ \ )$	10
$computeBMI(\ \ )$	10
$TestPersonWeight$ class (Compiles and runs)	
Instance of $PersonWeight(\ \ )$	4
Prompt and capture of input	16
Instance of $PersonWeight(\ldots)$	5
Printing person’s weight info to command prompt:	15
Correct output	10
	100

Sample Outputs:



```
Enter person's name: John Doe
Enter person's year of birth: 1990
Enter person's height in meters: 1.36
Enter person's weight in kilograms: 100.9
     Full Name: John Doe
           Age: 30
        Height: 1.36
        Weight: 100.90
Classification: Obese
```

Figure 1: Sample Run 1



```
Enter person's name: Jane Jack
Enter person's year of birth: 2000
Enter person's height in meters: 1.1
Enter person's weight in kilograms: 30.2
     Full Name: Jane Jack
           Age: 20
        Height: 1.10
        Weight: 30.20
Classification: Normal Weight
```

Figure 2: Sample Run 2