Assignment Report for [Assignment 03]

Course and Section	[CSC].[215]
Assignment Name	[Assignment 03]
Due Date and Time	[March 1] @ [11:59pm]
First Name and Last Name	Arjun Parujanwala
SFSU Email Account	[aparujanwala@sfsu.edu]
First Name and Last Name of Teammate	[Troy] [Mueller]
SFSU Email Account of Teammate	[tmueller@sfsu.edu]

PART A

Question Description and Analysis:

This part of the assignment asks that…create a BMI Calculator

Answer:

This is my answer…

1. The problem wants us to create a BMI calculator program in which the user can enter the user's weight and height. The program will then calculate if the user is underweight, healthy, or overweight. It will display that calculation with a scale giving a good view on where the user is at with their health. The requirements of the program must have an English and Metric version in which the user can choose which one they desire. It must also be able to calculate the calculation no matter if it is in centimeters or inches. We organize the solution by creating methods in order to target each important problem we must solve. The important elements are asking for the user their name, height and weight. And of course calculating the BMI. After that the program must display the result and an ending message to the user. When coding the solution, I must pay close attention to the

organization and names of the code in order to solve problems much easier and communicate better with the grader and interviewer in the future.

2. Done

1.

   a. public static double getWeight(String version)

   b. public is the component meaning that it is accessible in all packages in the program.

   c. getWeight(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.

   d. getWeight(String version) is the method signature. A method signature is the structure of the method that is designed by the programmer.

   e. The parameter “version” contains the exact parameter “english”

   f. String version is the formal parameter

   g. The caller of the method is “if (version.equals(“english”) and it is also the actual parameter.

   h. The one main task that the body does is ask the user for their height in double value.

   i. The method returns the user input of their height. EX: 71

   j. The return type is a double as it ask for nextDouble();

The reason why this method is in the program is because it completes an important step to run the program. It takes in the user’s input for weight so we can then use it later when the getBMI wants the number in order to calculate the BMI.

2.

   a. public static double getHeight(String version)

   b. public is the component meaning that it is accessible in all packages in the program.

c. getHeight(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.

d. getHeight(String version) is the method signature. A method signature is the structure of the method that is designed by the programmer.

e. The parameter “version” contains the exact parameter “english”

f. String version is the formal parameter

g. The caller of the method is “if (version.equals(“english”) and it is also the actual parameter.

h. The one main task that the body does is ask the user for their height in double value.

i. The method returns the user input of their weight. EX: 130.0

j. The return type is a double as it ask for nextDouble();

The reason why this method is in the program is because it completes an important step to run the program. It takes in the user’s input for height so we can then use it later when the getBMI wants the number in order to calculate the BMI.

3.

a. public static double getBMI(String version, double height, double weight);

b. public is the component meaning that it is accessible in all packages in the program.

c. getBMI(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.

d. getBMI(String version, double height, double weight) is the method signature. A method signature is the structure of the method that is designed by the programmer.

e. The parameter “version” contains the exact parameter “english.” The parameters “double height and double weight” contain doubles that depend on what the user inputs.

f. String version, double height, double weight are the formal parameters.

g. the callers of this method are if (version.equals("english")), bmi = (weight/(height * height)) * 703;, bmi = weight / ((height/100) * (height/100));.

h. The main task that this method does is calculate the bmi depending on the version you chose.

i. The method returns the bmi of the user rounded to the hundredths.

j. The return type is a double.

This is a crucial part of the program because if quite literally completed the entire objective to get the BMI of the user. This method only has 1 call when generateAndDisplayBMITable uses it in order to make a list of BMI.

4.

a. public static string summary Report(double bmi, String name)

b. public is the component meaning that it is accessible in all packages in the program.

c. summaryReport(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.

d. summaryReport(double bmi, String name) is the method signature. A method signature is the structure of the method that is designed by the programmer.

e. The "double bmi" contains the exact parameter bmi (depends) and "String name" contains the exact parameter of the user's name.

f. double bmi, String name are the formal parameters

g. The callers of this method are System.out.println("\n– SUMMARY REPORT for " + name.toUpperCase()); and System.out.printf("-- Weight Status: %s%n", status);

h. The main task of this method body is to gather your bmi , name and weight status then give a summary to the user.
i. the method returns the weight status
j. the return statement returns a string.

This is the summary report and it is important for the program because it displays the current user’s BMI and weight status.

5.

a. public static void generateAndDisplayBMITable(String version, double lowWeight, double highWeight, double height, initialWeight, double initialBMI)
b. public is the component meaning that it is accessible in all packages in the program.
c. generateAndDisplayBMITable(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.
d. public static void generateAndDisplayBMITable(String version, double lowWeight, double highWeight, double height, double, initialWeight, double initialBMI) is the method signature. A method signature is the structure of the method that is designed by the programmer.
e. String version contains “english,” double lowWeight, double highWeight, double height, initialWeight contains user-inputted numbers. double initialBMI contains a calculated number.
f. String version, double lowWeight, double highWeight, double height, initialWeight, double initialBMI are the formal parameters.
g. Pretty much, the entire code is a caller except the styling.

h. The main task of the body is to make an entire list from your lowWeight to highWeight and calculate bmi for each and everyone and also determine the weight status.

i. Code does not return anything

j. No return type.

This step completes the last step and is very important as it shows the user their BMI at different weight levels. This has only 1 call in each version.

3. The program runs and compiles very well. It is able to take the desired metric or english measurements from the user and display a calculation in an easy to understand way friendly to the user. What works is of course getting the right BMI calculation from the height and weight. Everything works well. A good plan to improve the program is to give more precise comments and organization for the interviewer.

Screenshots of Outputs and Explanation:

These screenshots show what I accomplished…

PART B

Question Description and Analysis:

This part of the assignment asks that…

Answer:

This is my answer…

1. This version of the program is different from the English version of it as it obviously used the metric system instead of the imperial system that the United States uses meaning that

instead of feet, inches and pounds, the metric system will be using centimeters and kilograms. The client may be from a different place that uses a different system of measurement and will want to input height and weight in different units thus this is a good solution for that in which the program can have two versions to use from. The question is asking for the program to be able to grab these different units of measurements from the user but still be able to calculate and output the right BMI for the user. The requirements of this will be asking the user to choose from English to Metric system and still be able to move on with the program even if the user puts in the wrong spelling. Important details observed from the desired outputs is that it must be in a easy to read format displaying the scale of where the user is at from overweight, underweight and healthy with different weights to show what is recommended to be healthy.

3.

1.

a. public static double getWeight(String version)

b. public is the component meaning that it is accessible in all packages in the program.

c. getWeight(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.

d. getWeight(String version) is the method signature. A method signature is the structure of the method that is designed by the programmer.

e. The parameter "version" contains the exact parameter "english"

f. String version is the formal parameter

g. The caller of the method is "if (version.equals("english") and it is also the actual parameter.

h. The one main task that the body does is ask the user for their height in double value.

i. The method returns the user input of their height. EX: 71

j. The return type is a double as it ask for nextDouble();

The reason why this method is in the program is because it completes an important step to run the program. It takes in the user's input for weight so we can then use it later when the getBMI wants the number in order to calculate the BMI.

2.

a. public static double getHeight(String version)

b. public is the component meaning that it is accessible in all packages in the program.

c. getHeight(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.

d. getHeight(String version) is the method signature. A method signature is the structure of the method that is designed by the programmer.

e. The parameter "version" contains the exact parameter "english"

f. String version is the formal parameter

g. The caller of the method is "if (version.equals("english") and it is also the actual parameter.

h. The one main task that the body does is ask the user for their height in double value.

i. The method returns the user input of their weight. EX: 130.0

j. The return type is a double as it ask for nextDouble();

The reason why this method is in the program is because it completes an important step to run the program. It takes in the user's input for height so we can then use it later when the getBMI wants the number in order to calculate the BMI.

3.

a. public static double getBMI(String version, double height, double weight);
b. public is the component meaning that it is accessible in all packages in the program.
c. getBMI(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.
d. getBMI(String version, double height, double weight) is the method signature. A method signature is the structure of the method that is designed by the programmer.
e. The parameter "version" contains the exact parameter "english." The parameters "double height and double weight" contain doubles that depend on what the user inputs.
f. String version, double height, double weight are the formal parameters.
g. the callers of this method are if (version.equals("english")), bmi = (weight/(height * height)) * 703;, bmi = weight / ((height/100) * (height/100));.
h. The main task that this method does is calculate the bmi depending on the version you chose.
i. The method returns the bmi of the user rounded to the hundredths.
j. The return type is a double.

This is a crucial part of the program because if quite literally completed the entire objective to get the BMI of the user. This method only has 1 call when generateAndDisplayBMITable uses it in order to make a list of BMI.

4.

a. public static string summary Report(double bmi, String name)
b. public is the component meaning that it is accessible in all packages in the program.

c. summaryReport(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.
d. summaryReport(double bmi, String name) is the method signature. A method signature is the structure of the method that is designed by the programmer.
e. The "double bmi" contains the exact parameter bmi (depends) and "String name" contains the exact parameter of the user's name.
f. double bmi, String name are the formal parameters
g. The callers of this method are System.out.println("\n– SUMMARY REPORT for " + name.toUpperCase()); and System.out.printf("-- Weight Status: %s%n", status);
h. The main task of this method body is to gather your bmi , name and weight status then give a summary to the user.
i. the method returns the weight status
j. the return statement returns a string.

This is the summary report and it is important for the program because it displays the current user's BMI and weight status.

5.

a. public is the component meaning that it is accessible in all packages in the program.
b. generateAndDisplayBMITable(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.
c. public static void generateAndDisplayBMITable(String version, double lowWeight, double highWeight, double height, double, initialWeight, double initialBMI) is the

method signature. A method signature is the structure of the method that is designed by the programmer.

d. String version contains “english,” double lowWeight, double highWeight, double height, initialWeight contains user-inputted numbers. double initialBMI contains a calculated number.
e. String version, double lowWeight, double highWeight, double height, initialWeight, double initialBMI are the formal parameters.
f. Pretty much, the entire code is a caller except the styling.
g. The main task of the body is to make an entire list from your lowWeight to highWeight and calculate bmi for each and everyone and also determine the weight status.
h. Code does not return anything
i. No return type.

This step completes the last step and is very important as it shows the user their BMI at different weight levels. This has only 1 call in each version.

2. ad.**Screenshots of Outputs and Explanation**:

These screenshots show what I accomplished…

PART C

Question Description and Analysis:

This part of the assignment asks that…to create a BMI calculator.

Answer:

This is my answer…

1. The question is asking for the user to be able to choose whichever version they desire from English to Metric. The client will want to choose as they may be from another country in which they use different units of measurements. The requirements of the program wants a method with only one statement that we can call. It also requires the program to be able to choose the right version from the user even if the user types a misspelled name. Important details are outputting the right questions after each time the user inputs their height, weight and name. An important problem that must be solved is finding which version the user desires even if they put in a typo. A good solution is detecting if the first letter is an M or an E and choosing the version from those letters only. That way if there is a mistake, it will not matter for the program. Designing the solution in this stage of the program's development involves carefully seeing which problem is causing an error and defining what kind of error it is to be able to come up with the right solutions and see where to look for them. Must pay attention to the naming conventions and comments as well as seeing if all the requirements are met for the client's needs.
2. Done
3.

1.

   a. public static double getWeight(String version)

   b. public is the component meaning that it is accessible in all packages in the program.

c. getWeight(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.

d. getWeight(String version) is the method signature. A method signature is the structure of the method that is designed by the programmer.

e. The parameter “version” contains the exact parameter “english”

f. String version is the formal parameter

g. The caller of the method is “if (version.equals(“english”) and it is also the actual parameter.

h. The one main task that the body does is ask the user for their height in double value.

i. The method returns the user input of their height. EX: 71

j. The return type is a double as it ask for nextDouble();

The reason why this method is in the program is because it completes an important step to run the program. It takes in the user’s input for weight so we can then use it later when the getBMI wants the number in order to calculate the BMI.

2.

a. public static double getHeight(String version)

b. public is the component meaning that it is accessible in all packages in the program.

c. getHeight(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.

d. getHeight(String version) is the method signature. A method signature is the structure of the method that is designed by the programmer.

e. The parameter “version” contains the exact parameter “english”

f. String version is the formal parameter

g. The caller of the method is “if (version.equals(“english”) and it is also the actual parameter.

h. The one main task that the body does is ask the user for their height in double value.

i. The method returns the user input of their weight. EX: 130.0

j. The return type is a double as it ask for nextDouble();

The reason why this method is in the program is because it completes an important step to run the program. It takes in the user’s input for height so we can then use it later when the getBMI wants the number in order to calculate the BMI.

3.

a. public static double getBMI(String version, double height, double weight);

b. public is the component meaning that it is accessible in all packages in the program.

c. getBMI(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.

d. getBMI(String version, double height, double weight) is the method signature. A method signature is the structure of the method that is designed by the programmer.

e. The parameter “version” contains the exact parameter “english.” The parameters “double height and double weight” contain doubles that depend on what the user inputs.

f. String version, double height, double weight are the formal parameters.

g. the callers of this method are if (version.equals("english")), bmi = (weight/(height * height)) * 703;, bmi = weight / ((height/100) * (height/100));.

h. The main task that this method does is calculate the bmi depending on the version you chose.

i. The method returns the bmi of the user rounded to the hundredths.

j. The return type is a double.

This is a crucial part of the program because if quite literally completed the entire objective to get the BMI of the user. This method only has 1 call when generateAndDisplayBMITable uses it in order to make a list of BMI.

4.

a. public static string summary Report(double bmi, String name)

b. public is the component meaning that it is accessible in all packages in the program.

c. summaryReport(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.

d. summaryReport(double bmi, String name) is the method signature. A method signature is the structure of the method that is designed by the programmer.

e. The “double bmi” contains the exact parameter bmi (depends) and “String name” contains the exact parameter of the user’s name.

f. double bmi, String name are the formal parameters

g. The callers of this method are System.out.println(“\n– SUMMARY REPORT for “ + name.toUpperCase()); and System.out.printf(“-- Weight Status: %s%n”, status);

h. The main task of this method body is to gather your bmi , name and weight status then give a summary to the user.

i. the method returns the weight status

j. the return statement returns a string.

This is the summary report and it is important for the program because it displays the current user’s BMI and weight status.

5.

a. public is the component meaning that it is accessible in all packages in the program.

b. generateAndDisplayBMITable(). It is named properly because it fits the standard for a properly named method and it also simply tells us what the method does.

c. public static void generateAndDisplayBMITable(String version, double lowWeight, double highWeight, double height, double, initialWeight, double initialBMI) is the method signature. A method signature is the structure of the method that is designed by the programmer.

d. String version contains “english,” double lowWeight, double highWeight, double height, initialWeight contains user-inputted numbers. double initialBMI contains a calculated number.

e. String version, double lowWeight, double highWeight, double height, initialWeight, double initialBMI are the formal parameters.

f. Pretty much, the entire code is a caller except the styling.

g. The main task of the body is to make an entire list from your lowWeight to highWeight and calculate bmi for each and everyone and also determine the weight status.

h. Code does not return anything

i. No return type.

This step completes the last step and is very important as it shows the user their BMI at different weight levels. This has only 1 call in each version.

4. The program runs and compiles very well. It satisfied all the client’s needs in this problem. The program can detect which version the client wants even if there is a typo and displays the right calculations and text for the user to easily read. There are no

problems with the program as it runs well. The plan to improve the program is to create a better look for the user to easily read and find if they are healthy, underweight, or overweight. A more user-friendly program is the best plan to improve the program as the program already runs well and outputs the right calculations. More comments explaining what each part does can also be applied for people to understand much better. Another part of the program that is able to work is taking in all the data from the user such as height, weight and finding the right calculation to give for the BMI problem. And outputting those numbers into an easy to read scale for the user.

Screenshots of Outputs and Explanation:

These screenshots show what I accomplished…

PART D

Question Description and Analysis:

This part of the assignment asks that…to create a Semester-Completion Plan and Study Plan

Answer:

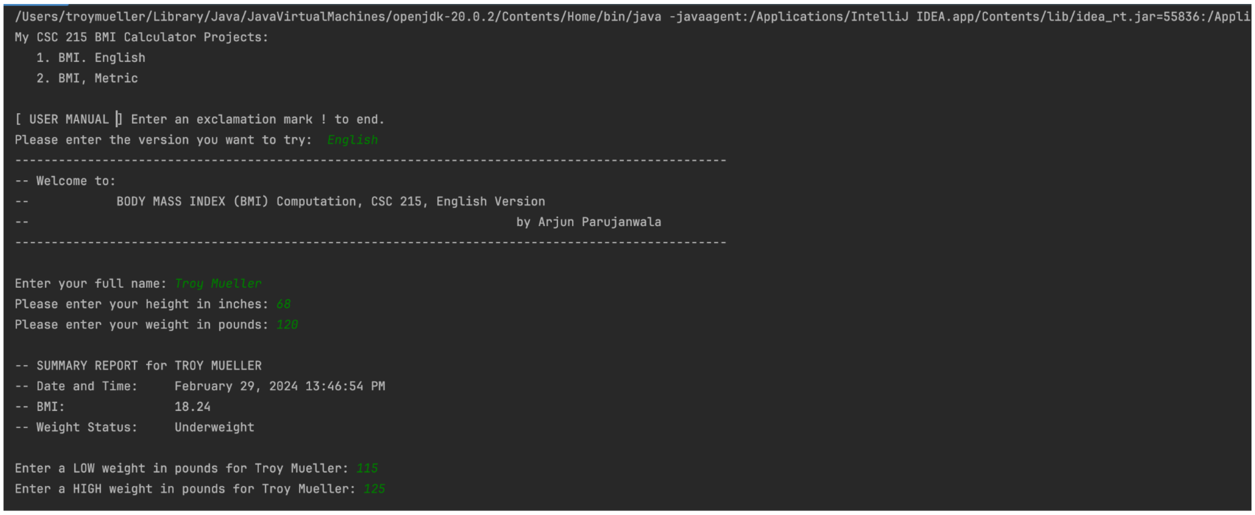
This is my answer…

Current goals for this course is of course to pass the course. But to stay on top of assignments and not procrastinate to the last minute as from experience, it destroys the quality of the work put in if it is rushed. To also find more resources outside of class to study up on and find new problems to solve in Java. Find more challenges to improve drastically. Youtube is a great resource that you will be looking into as there are hundreds of videos and tutorials on how to code anything. Currently doing well in this course and the goals are able to be achieved. Changes that must be made is to start working on assignments the minute they get released in order to manage time better and not be overwhelmed with all the work that must be made. Will implement these changes by doing them immediately and work hard to achieve the goals.

2. Will have an hour or 30 min session a day depending on the schedule working on Leetcode in order to be prepared for interviews and internships in the future.

Screenshots of Outputs and Explanation:

These screenshots show what I accomplished…



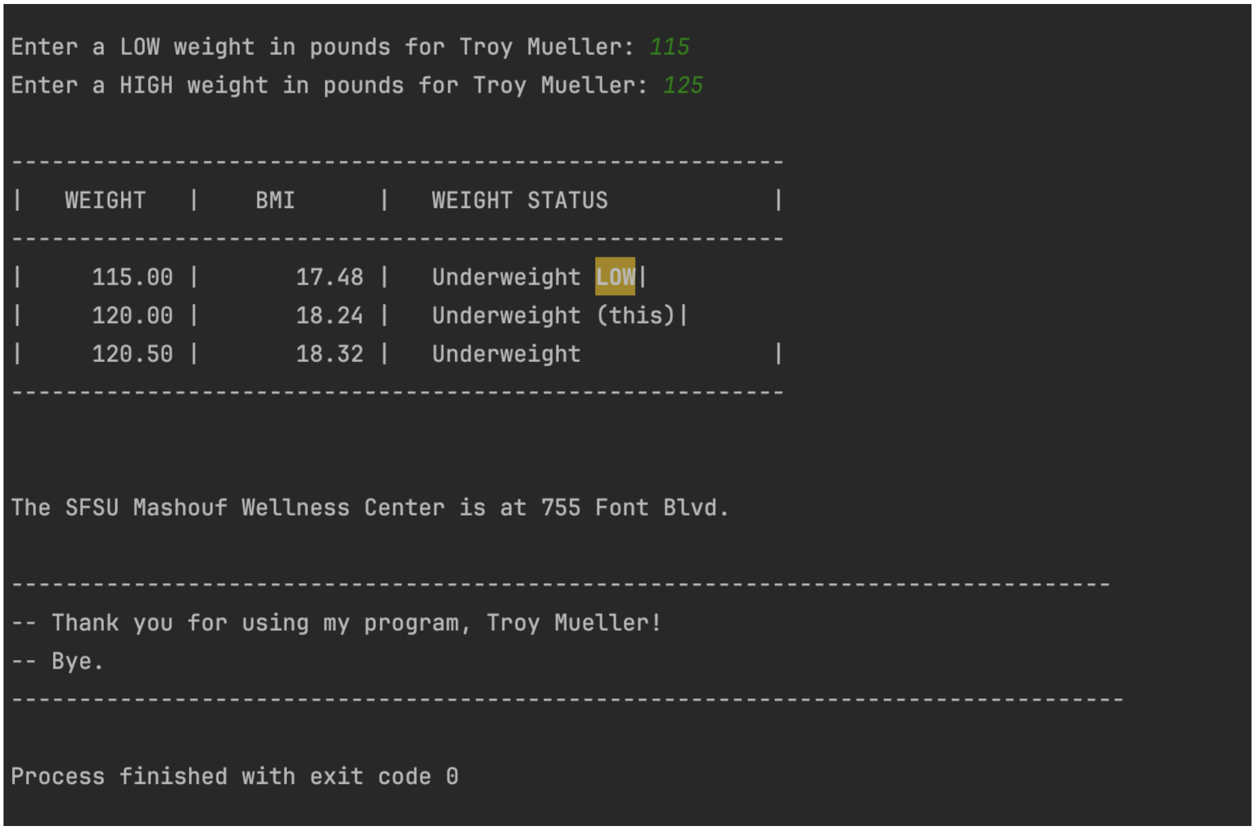
```
/Users/troymueller/Library/Java/JavaVirtualMachines/openjdk-20.0.2/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=55836:/Appli
My CSC 215 BMI Calculator Projects:
   1. BMI. English
   2. BMI, Metric

[ USER MANUAL ] Enter an exclamation mark ! to end.
Please enter the version you want to try:  English
--------------------------------------------------------------------------------------------------
-- Welcome to:
--            BODY MASS INDEX (BMI) Computation, CSC 215, English Version
--                                                                   by Arjun Parujanwala
--------------------------------------------------------------------------------------------------

Enter your full name: Troy Mueller
Please enter your height in inches: 68
Please enter your weight in pounds: 120

-- SUMMARY REPORT for TROY MUELLER
-- Date and Time:     February 29, 2024 13:46:54 PM
-- BMI:               18.24
-- Weight Status:     Underweight

Enter a LOW weight in pounds for Troy Mueller: 115
Enter a HIGH weight in pounds for Troy Mueller: 125
```



```
Enter a LOW weight in pounds for Troy Mueller: 115
Enter a HIGH weight in pounds for Troy Mueller: 125

---------------------------------------------------------
|   WEIGHT   |    BMI      |   WEIGHT STATUS            |
---------------------------------------------------------
|     115.00 |       17.48 |   Underweight LOW|
|     120.00 |       18.24 |   Underweight (this)|
|     120.50 |       18.32 |   Underweight              |
---------------------------------------------------------


The SFSU Mashouf Wellness Center is at 755 Font Blvd.

---------------------------------------------------------------------------------
-- Thank you for using my program, Troy Mueller!
-- Bye.
----------------------------------------------------------------------------------

Process finished with exit code 0
```