

Java1: Lesson 3 – Lab Project

1. Add **summary** at the top, **documentation** at the bottom and **comments** where necessary.
- Recordings and interviews are an exception, but they should be included in your submission code.
2. **Cite** the source of your information.
3. Refrain from including anything **you don't fully understand**.

2

100 points

2-1. Analyze the code below using the provided test data.

1. Insert line numbers into the code: <https://bit.ly/3DMjFxN>
2. Submit your analysis in .docx format.

2-2. Update the code to take input in **kg/cm** and output in **pound/feet/inch**.

1. Apply your coding style.
2. Add **comments** to explain the code.
3. Test the updated code using your own test data.

```
System.out.print("Enter weight in pounds: ");
double weight = input.nextDouble();           //weight=50
System.out.print("Enter height in inches: ");
double height = input.nextDouble();           //height=100
final double KILOGRAMS_PER_POUND = 0.4535;
final double METERS_PER_INCH = 0.0254;
double weightInKilograms = weight * KILOGRAMS_PER_POUND;
double heightInMeters = height * METERS_PER_INCH;
double bmi = weightInKilograms / (heightInMeters * heightInMeters);
System.out.println("BMI is " + bmi);
if (bmi < 18.5)
    System.out.println("Underweight");
else if (bmi < 25)
    System.out.println("Normal");
else if (bmi < 30)
    System.out.println("Overweight");
else
    System.out.println("Obese");
//your output:
```

Sample output	Weight (lb): 220 Height(ft): 6 Height(in): 1 ===== Your BMI is *** which is Normal
---------------	--

File name	J103_2.docx J103_2.java, J103_2.png
-----------	--

3

100 points

Analyze, trace and interpret line by line the following code snippets using sample data.

1. Fill in any missing information (***)
2. Submit your analysis in .docx format.

3-1. This code is ***

```
1 if (number1 < number2) {
2     int temp = number1;
3     number1 = number2;
4     number2 = temp;
5 }
```

3-2. Translate the given logical expression into plain language and then fill in the placeholder.

```
if (N % 3 == 0 && N % 5 == 0)
```

→ if N ***

and the statement is true when N = ***

3-2. Convert each Boolean condition into words and fill in the placeholder.

```
1 if ((N % 4) == 0) {                // 1st if range: line *** ~ line ***
2     if ((N % 100) == 0) {          // 2nd if range: line *** ~ line ***
3         if ((N % 400) == 0) {      // 3rd if range: line *** ~ line ***
4             . . . // when N is ***
5         }
6     } else {                       // else range: line *** ~ line ***
7         . . . // when N is ***
8     }
9 }
```

3-3. Translate the given logical expression into plain language and then fill in the placeholder.

```
if (N % 4 == 0 && N % 100 != 0) || (N % 400 == 0)
```

→ if N ***

and the statement is true when N = ***

3-4. Compare the algorithm in 3-2. and 3-3.

proceed to the next page →

3-5. @

```
1 int A1 = A / 10; // A1 is ***
2 int A2 = A % 10; // A2 is ***
3 int B1 = B / 10; // B1 is ***
4 int B2 = B % 10; // B2 is ***
5 if (A == B)
6     . . .
7 else if (B2 == A1 && B1 == B2)
8     . . . //when ***
9 else if (B1 == A1 || B1 == A2 || B2 == A1 || B2 == A2)
10    . . . //when ***
11 else
12    . . . //when ***
```

File
name

J103_3.docx