Assignment 3 Instructions

1. Assignment 01: **50 points total with 2 E.C. points (For class participation, for extra work helping others in class, for not being late on submitting your assignment.)**
2. Due Date & Time: **09-21-2020 at 11:59 PM**

WHAT TO SUBMIT

Submit 4 files to iLearn by the deadline. [48pts + 2 E.C. pts = 50 points]

- 2 Files: Please submit 2 files to iLearn: **TableBmi.java**, **TableBmiPro.java**, <YourOwnIdea>.java. **[40 points]**
- 1 File: Submit 1 Word/PDF file which is a filled-out, downloaded local copy of this Google page on your local computer, named "firstname-lastname-assignment-3-report.pdf". Fill this out with screenshots and your reflection, then save it as Word or PDF **[8 points]**

HOW TO SUBMIT

Please upload all 3 files separately via iLearn Assignments Submission

---

GUIDELINES FOR ALL ASSIGNMENTS:

1. Each assignment includes a code portion and a non-code portion. Please submit both 2 portions.
   a. Code portion: Your source code files, only the files which you create and edit.
   b. Non-code portion: Your assignment report, only 1 **Word** or **PDF** file.

2. Please submit all required files separately, un-zipped, via iLearn Assignments Submission

3. Always **read through the entire assignment before starting and submitting any of it. Missing files or missing requirements will result in deducted points**

Assignment 3

BODY MASS INDEX (BMI) COMPUTATION PRO

❒ Part 1: BMI History Pro [18 points]

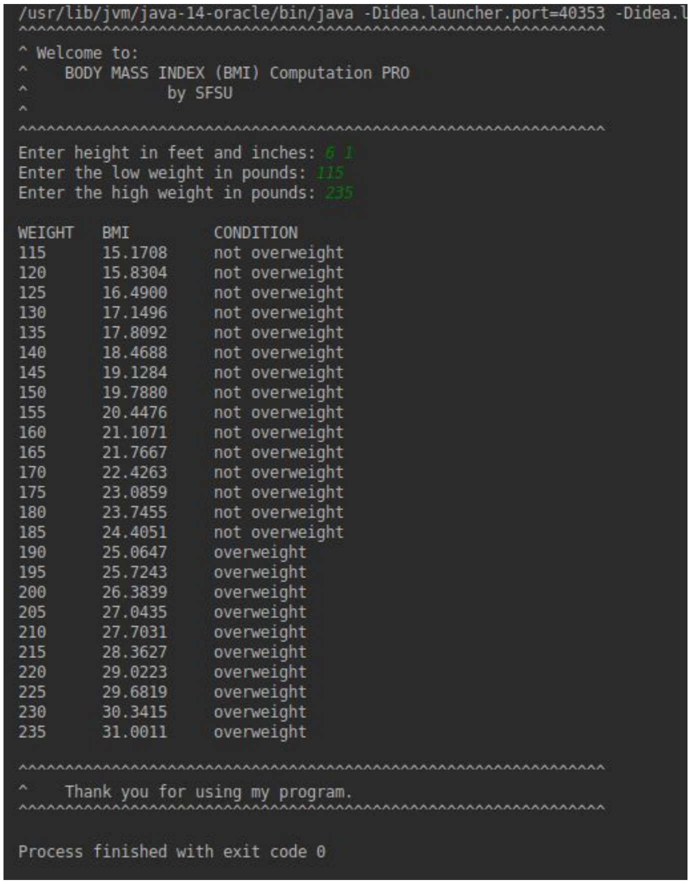
1. Prompt our user to enter his/her height in feet and inches (two integers).
2. Prompt our user to enter his/her **lowest weight** in pounds (an integer).
3. Prompt our user to enter his/her **heaviest weight** in pounds (an integer).
4. Print a table of Body Mass Index (BMI) for the height entered:

a) Weights range from the low weight to the high weight, at increments of 5 pounds.
   - This means to get more than 1 line low weight and high weight should have more than 5 pounds of difference
   - To get additional decimals behind decimal point, you may want to cast one of the variables into a float or a double
   - prompt user twice int user2 = scan.nextInt()// feet
   - int user3 = scan.nextInt()// inches
   - 703 * weight(height * height) // height is in inches Math.pow(heigh, 2)

b) Each row of the table lists
   - The value of WEIGHT (an integer), followed by spaces, then
   - The value of BMI to four decimal places (a float), followed by spaces, then
   - The CONDITION whether overweight (BMI > 25), or not overweight (BMI <= 25).

5. Document our code carefully. Our program output must be **<u>identical</u>** to the sample output (except author name).

OUTPUT OF SAMPLE RUN FOR PART 1



```
/usr/lib/jvm/java-14-oracle/bin/java -Didea.launcher.port=40353 -Didea.l
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^ Welcome to:
^    BODY MASS INDEX (BMI) Computation PRO
^               by SFSU
^
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Enter height in feet and inches: 6 1
Enter the low weight in pounds: 115
Enter the high weight in pounds: 235

WEIGHT   BMI         CONDITION
115      15.1708     not overweight
120      15.8304     not overweight
125      16.4900     not overweight
130      17.1496     not overweight
135      17.8092     not overweight
140      18.4688     not overweight
145      19.1284     not overweight
150      19.7880     not overweight
155      20.4476     not overweight
160      21.1071     not overweight
165      21.7667     not overweight
170      22.4263     not overweight
175      23.0859     not overweight
180      23.7455     not overweight
185      24.4051     not overweight
190      25.0647     overweight
195      25.7243     overweight
200      26.3839     overweight
205      27.0435     overweight
210      27.7031     overweight
215      28.3627     overweight
220      29.0223     overweight
225      29.6819     overweight
230      30.3415     overweight
235      31.0011     overweight

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^    Thank you for using my program.
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Process finished with exit code 0
```

```
for (...
        System.out.print(“^”);
}
System.out.print(“^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^”);
```

SUBMISSION INSTRUCTIONS

1. Submit the 1 **TableBmi.java** file directly on iLearn

2. Take a screenshot of the output of your program and paste it here

Output :

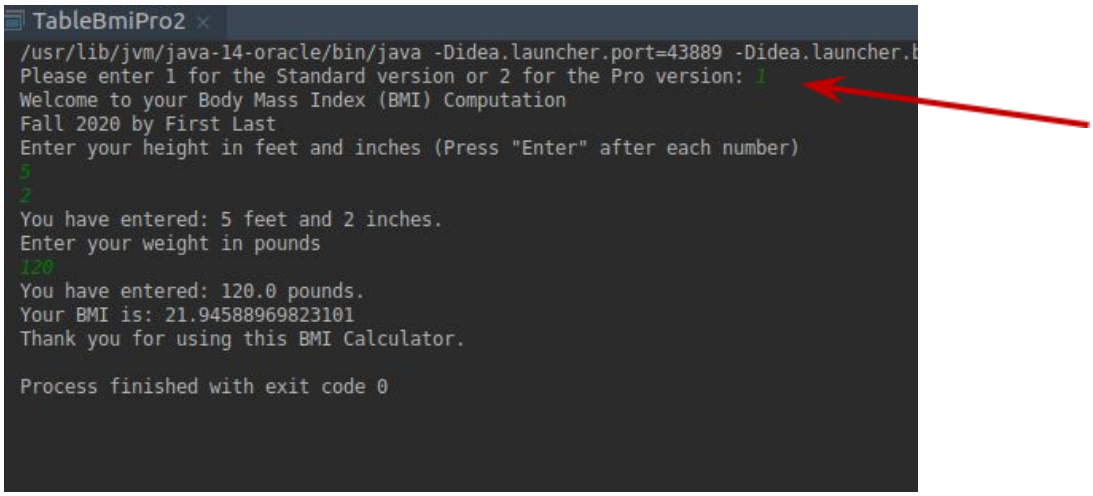
Run: TableBmiPro

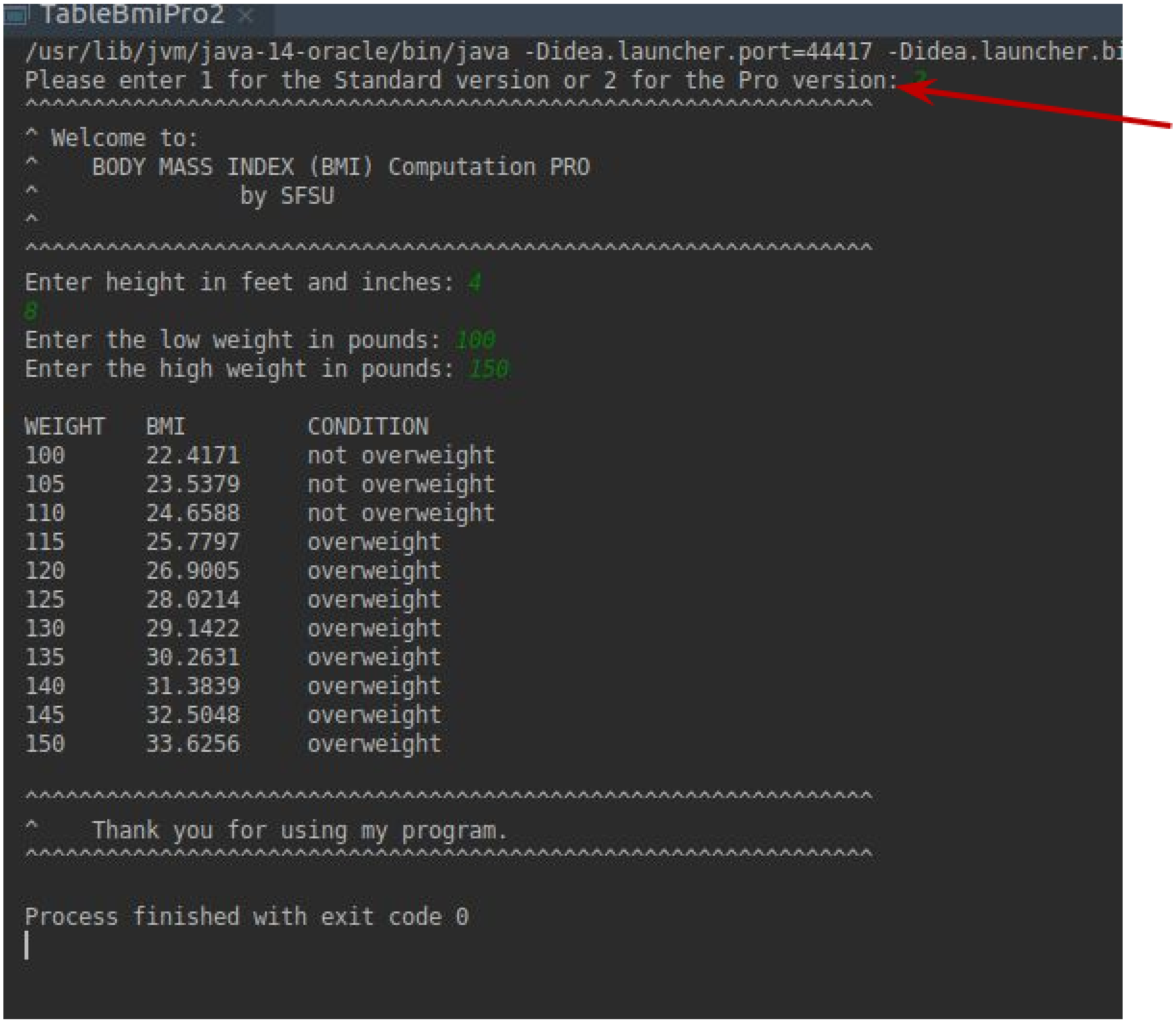
```
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:D:\App
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^ Welcome to:
^    BODY MASS INDEX (BMI) Computation PRO
^               by Mos
^
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Enter height in feet and inches: 6 1
Enter the low weight in pounds: 115
Enter the high weight in pounds: 235

WEIGHT   BMI         CONDITION
115      15.1708     not overweight
120      15.8304     not overweight
125      16.4900     not overweight
130      17.1496     not overweight
135      17.8092     not overweight
140      18.4688     not overweight
145      19.1284     not overweight
150      19.7880     not overweight
155      20.4476     not overweight
160      21.1071     not overweight
165      21.7667     not overweight
170      22.4263     not overweight
175      23.0859     not overweight
180      23.7455     not overweight
185      24.4051     not overweight
190      25.0647     overweight
195      25.7243     overweight
200      26.3839     overweight
205      27.0435     overweight
210      27.7031     overweight
215      28.3627     overweight
220      29.0223     overweight
225      29.6819     overweight
230      30.3415     overweight
235      31.0011     overweight

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^    Thank you for using my program.
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Process finished with exit code 0
```

❒ Part 2: BMI Code Reuse [10 points]

1. So far, we have written 2 versions of the BMI program, BMI Standard version from assignment 1 and BMI History Pro version from this Assignment 2, part 1. Reusing the programs, please make our BMI offer users to choose which version they want to use. (After users make their choice, they should be able to use the program they chose.)
2. Document our code carefully using the provided guidelines. Include screenshots and notes in our report.

OUTPUT OF SAMPLE RUNS FOR PART 2





```
TableBmiPro2
/usr/lib/jvm/java-14-oracle/bin/java -Didea.launcher.port=44417 -Didea.launcher.bi
Please enter 1 for the Standard version or 2 for the Pro version: 2
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^ Welcome to:
^    BODY MASS INDEX (BMI) Computation PRO
^               by SFSU
^
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Enter height in feet and inches: 4
8
Enter the low weight in pounds: 100
Enter the high weight in pounds: 150

WEIGHT   BMI         CONDITION
100      22.4171     not overweight
105      23.5379     not overweight
110      24.6588     not overweight
115      25.7797     overweight
120      26.9005     overweight
125      28.0214     overweight
130      29.1422     overweight
135      30.2631     overweight
140      31.3839     overweight
145      32.5048     overweight
150      33.6256     overweight

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^    Thank you for using my program.
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Process finished with exit code 0
```

SUBMISSION INSTRUCTIONS

1. Submit the 1 **TableBmiPro.java** file directly on iLearn

2. Take a screenshot of the output of your program for option 1 and option 2 and paste it here

Output for Option 1 :

```
Run - CSC210Asmt03
Run:  TableBmi  TableBmiPro
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:D:\Applications\JetBrains\I
Please enter 1 for the Standard version or 2 for the Pro version: 1
============================
Welcome to my BMI calculator
============================
Please enter your height in feet and inches (press Enter after each number)
Feet: 5
Inches: 2
Your height: 5'2"

Please enter your weight in pounds: 120
Your weight: 120.00 lbs

>>> Your BMI is: 21.95
Process finished with exit code 0
```

Output for Option 2:

```
Run - CSC210Asmt03
Run:  TableBmi   TableBmiPro

"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:D:\Applications\JetBrains\I
Please enter 1 for the Standard version or 2 for the Pro version: 2
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^ Welcome to:
^    BODY MASS INDEX (BMI) Computation PRO
^               by Mos
^
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Enter height in feet and inches: 4
8
Enter the low weight in pounds: 100
Enter the high weight in pounds: 150

WEIGHT   BMI         CONDITION
100      22.4171     not overweight
105      23.5379     not overweight
110      24.6588     not overweight
115      25.7797     overweight
120      26.9005     overweight
125      28.0214     overweight
130      29.1422     overweight
135      30.2631     overweight
140      31.3839     overweight
145      32.5048     overweight
150      33.6256     overweight

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^    Thank you for using my program.
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Process finished with exit code 0
```

❒Part 3: Your own idea [10 points]

It can be similar to part 1 or 2.

Please use what you learned from Loops.

a. In Assignment 1, Part 3 and in Assignment 2, Part 2: Your own idea, you created your own idea of a program. For this assignment, you must improve upon your first program by adding what you learned from Loops

b. Must include Loops, using either "for", or "while" or "do while"

c. And, must use the loop to output more than 1 line of result

d. Must provide an explanation to the user of your idea "This program will help you... " (2 points)

FILE NAME REQUIREMENTS

<YourFileName>.Java

SUBMISSION INSTRUCTIONS

1. Submit the 1 <YourFileName>.Java file directly on iLearn

2. Take a screenshot of the output of your program run for **all conditions** and paste it here

Explanation:
This program calculates the compound interest rate and displays the generated interests and the cumulative balances as a table. The execution loop is as follows:

- The user enters the required parameters to calculate the periodic compounding interest. These are:
  - the principal deposit amount,
  - the compounding frequency,
  - the nominal annual interest rate, and
  - the deposit term.
- The program prints and re-summarizes these parameters to the user.
- The program will determine whether to display a yearly or a quarterly report. This depends on the deposit term length.
- These parameters are then used to calculate the interests generated and subsequently, the new

balance for each period.

- These results are printed in each row. Each row will present the following information:
  - the deposit length,
  - the interest generated at that period,
  - the cumulative interest generated,
  - and the new principal balance at that period
- Depending on the deposit term length, the number of rows may vary. For example:
  - a yearly report for a six-year term deposit will output seven rows, one row for each year, with the first row at year 0
  - a quarterly report for a two-year deposit will output nine rows, one row for each quarter, with the first row at year 0

- After the table is displayed, the user is then presented with a summary, presenting:
  - their original principal sum (what they started with)
  - the total generated interest (how much they made)
  - and their final balance (the total at the end)

There are infinite possible outcomes for this program.
However, there are two main ways the output may differ: the report may be on a **quarterly** or **yearly** basis. The program will generate a quarterly report if the deposit term is 2 years or less (a yearly report for a one- or two-year deposit would be pretty useless). Otherwise, it will generate a yearly report.

Note on output formatting

- **"Year 0"**
  - The table starts at year 0 deliberately in order to illustrate the starting point of the deposit. This is meant to contrast the starting interest and how it increases over the period.
- **Deposit length column decimal years**
  - The years are formatted to contain two decimal points for quarterly reports.
- **Rounding**
  - All dollar amounts are rounded to the nearest cent.

The two example cases are on the following pages.

Example 1:

The user inputs a deposit term that is twelve years.

The program will output a yearly report displaying the interests and principal balance at the end of each period.

```
Run - CSC210Asmt03
Run: InterestCalculator
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:D:\Applications\JetBrains\I
COMPOUND INTEREST CALCULATOR
============================
Enter your principal deposit amount (in $): 20000
Enter your compounding frequency (per annum): 4
Enter your compound annual interest rate (in %): 4.8
Enter your deposit term (in years): 12

=========================================================
Your yearly compound interest report:
Principal sum of $20000.00 with a 12.00-year deposit term
compounding 4 time(s) per year at 4.80% p.a.
=========================================================
 Deposit |     Interest |   Cumulative |          New
  length |    at period |     interest |    principal
---------------------------------------------------------
 0.00 yr | $       0.00 | $       0.00 | $  20,000.00
 1.00 yr | $     977.42 | $     977.42 | $  20,977.42
 2.00 yr | $   1,025.19 | $   2,002.60 | $  22,002.60
 3.00 yr | $   1,075.29 | $   3,077.89 | $  23,077.89
 4.00 yr | $   1,127.84 | $   4,205.73 | $  24,205.73
 5.00 yr | $   1,182.96 | $   5,388.69 | $  25,388.69
 6.00 yr | $   1,240.77 | $   6,629.46 | $  26,629.46
 7.00 yr | $   1,301.41 | $   7,930.86 | $  27,930.86
 8.00 yr | $   1,365.01 | $   9,295.87 | $  29,295.87
 9.00 yr | $   1,431.72 | $  10,727.59 | $  30,727.59
10.00 yr | $   1,501.69 | $  12,229.27 | $  32,229.27
11.00 yr | $   1,575.07 | $  13,804.35 | $  33,804.35
12.00 yr | $   1,652.05 | $  15,456.40 | $  35,456.40
=========================================================
 Original principal sum: $  20,000.00
     Interest generated: $  15,456.40
Final principal balance: $  35,456.40

Process finished with exit code 0
```

Example 2:

The user inputs a deposit term that is two years.

The program will output a quarterly report displaying the interests and principal balance at the end of each period.

```
Run - CSC210Asmt03
Run: InterestCalculator
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:D:\Applications\JetBrains\I
COMPOUND INTEREST CALCULATOR
============================
Enter your principal deposit amount (in $): 10000
Enter your compounding frequency (per annum): 4
Enter your compound annual interest rate (in %): 5
Enter your deposit term (in years): 2

=========================================================
Your quarterly compound interest report:
Principal sum of $10000.00 with a 2.00-year deposit term
compounding 4 time(s) per year at 5.00% p.a.
=========================================================
 Deposit |     Interest |   Cumulative |          New
  length |    at period |     interest |    principal
---------------------------------------------------------
 0.00 yr | $       0.00 | $       0.00 | $  10,000.00
 0.25 yr | $     125.00 | $     125.00 | $  10,125.00
 0.50 yr | $     126.56 | $     251.56 | $  10,251.56
 0.75 yr | $     128.14 | $     379.71 | $  10,379.71
 1.00 yr | $     129.75 | $     509.45 | $  10,509.45
 1.25 yr | $     131.37 | $     640.82 | $  10,640.82
 1.50 yr | $     133.01 | $     773.83 | $  10,773.83
 1.75 yr | $     134.67 | $     908.50 | $  10,908.50
 2.00 yr | $     136.36 | $   1,044.86 | $  11,044.86
=========================================================
 Original principal sum: $  10,000.00
     Interest generated: $   1,044.86
Final principal balance: $  11,044.86

Process finished with exit code 0
```

What a rate!

❒ Part 4: Comment your code [2 points]

Every Java file you write in this assignment will require you to include descriptive comments.

In this assignment, you are tasked with writing a descriptive

1. Headers
2. Comments

You can write comments in two ways:

- Single-line comments using the // notation.
- Multi-line comments using the /* and */ notation.

a. Include a proper header at the top of every Java file. Figure 1

Header Format
/* * Assignment **<assignment number>** * Description: **<program description>** * Name: **<your name>** * ID: **<your SFSU ID number>** * Class: CSC 210-**<section number>** * Semester: **<current semester>** */

Replace each tag (such as **<assignment number>**) with the appropriate text.

You should adhere to this format as closely as possible. You do not need to include the **<>** symbols in your header fields.

b. Only if you work with a Study Buddy, include your Buddy's name in your header at the top of every Java file. Figure 1

Header Format
/* * Assignment **<assignment number>** * Description: **<program description>** * Name: **<your name>** * Teammate: **<Study Buddy name>** * ID: **<your SFSU ID number>** * Class: CSC 210-**<section number>** * Semester: **<current semester>** */

Replace each tag (such as **<assignment number>**) with the appropriate text.

You should adhere to this format as closely as possible. You do not need to include the **<>** symbols in your header fields.

Figure 1

```
assignment1.java   KnockKnock.java
/*
 * Assignment <assignment number>
 * Description: <program description>
 * Name: <your name>
 * ID: <your SFSU ID number>
 * Class: CSC 210-<section number>
 * Semester: <current semester>
 */
```

b. Place your comments at the top of **each** Statement, however you don't need to comment print (i.e. anything that starts with System.out.print…) statements. An example of commenting codes is included below in Figure 2:

Figure 2

```
//To create a scanner object
Scanner scan = new Scanner(System.in);
```

❒Part 5: Reflection 50 words [8 points]

*Points will be deducted for less than 50 words.

Please put what was helpful and what was not helpful in working on this program.

And also please tell me how you would improve your own program from part 3, if you were given more time.

Only if you work with a Study Buddy, write down how your buddy helped you, i.e. Don helped me learn a new technique for understanding OOP, by thinking about containers.

Place your reflection in this Google Form

The reflection below is identical to the one submitted on Google Forms.

The instructions were clear for the most part. There was a slight ambiguity on how part two were to be completed, but that was later clarified in the Slack group chat. Perhaps hints on how to use string formatting methods, such as *printf* and *String.format()* may be useful for those who have never used it. If I had more time, I would validate the user's input and put each input prompt through an infinite loop until the input is satisfied.

I taught Amber on how to use *printf* and how she would be able to format the output in a table-like format. Through this, she in turn, taught me that there were shortcuts you could use for "System.out.println()"! Simply typing "sout" and pressing tab would auto-complete the whole thing.