Project 3 Instructions [10 points]

1. Due Date & Time: **October 7th, 2022 at 11:59 pm (PT)**

2. What to submit: Submit 1 zip file containing 3 files below to iLearn by the deadline.

   - 2 JAVA Files: TableBmiPro.java [4 points], DiceRoll2.java [5 points]
   - 1 File: Make a document that shows the screen captures of execution of your programs and learning points in Word or PDF. Please make sure you capture at least 2 executions for each of the programs (total of 4 screen captures) and write one paragraph reflecting on what you learned from this exercise [1 point]

   Please submit all required files together in a zip file, via iLearn Assignments Submission
   Please make the zip file name according to the naming convention: proj3_<FIRST NAME>_<LAST NAME>.zip

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

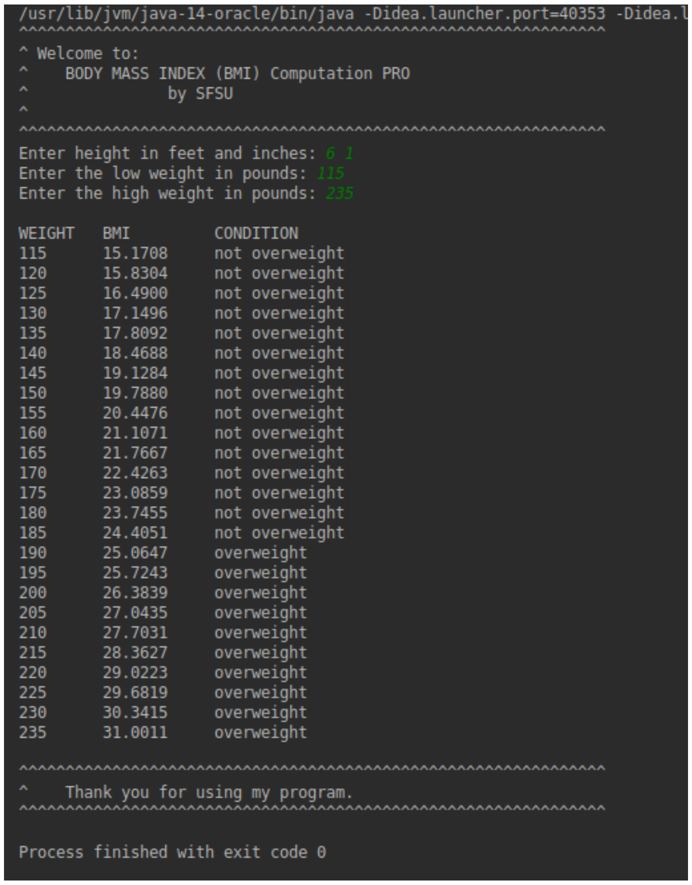
---

1. BMI History Pro

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.
      - To get multiple lines of output, low weight and high weight should have more than 5 pounds of difference
      - To get decimal places in BMI values, you may want to cast one of the variables into a float or a double

   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 your code carefully. Your program output must be **identical** to the sample output (except author name).
6. BMI Information:
   https://www.cdc.gov/nccdphp/dnpao/growthcharts/training/bmiage/page5_2.html
   https://www.cdc.gov/obesity/adult/defining.html

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
```

2. DiceRoll2.java

Make a dice and roll it n times. In the examples below, n = 100 or n = 10000. n is an user input and it can range between 1 to 1 million.
In each run, you have to count how many times each face [1-6] appears, print the probability of each occurrence to check if the random number actually works.

Tip: During the early stage of your code development, make your program roll once or 10 times. It will make it easier to debug if there is an issue. If you run your program 100000 times then you will have to wait a long time! After you have ensured that your program works, then you can increase the number of rolls to be 100, or 200 or 10000.

Remember, you have to ask the user to enter the value of n. Do NOT hard-code the values yourself in the final program.

Comment your code carefully assuming you are teaching your friend to learn from your comments.
Include screenshots and notes in your report.

OUTPUT OF SAMPLE RUNS FOR PART 2

```
[When the number of rolls is 100]
Occurrence of each face is:
11, 20, 22, 13, 19, 15: 100

Therefore, the probability of each face is:
0.11, 0.2, 0.22, 0.13, 0.19, 0.15


[When the number of rolls is 10000]
Occurrence of each face is:
1667, 1700, 1665, 1683, 1616, 1669: 10000

Therefore, the probability of each face is:
0.1667, 0.17, 0.1665, 0.1683, 0.1616, 0.1669
```

3. You also need to create a Word or PDF file that contains:

1. 2 screen captures for each of the problems executed above.
2. Reflection (1 point):

   Please write 200 words or more about what you learned, what challenges you faced, and how you solved it. You can also write about what was most frustrating and what was rewarding. When you write about what you learned, please be specific and list all the new terms or ideas that you learned!

<u>Now, create a zip file with the 2 Java files and the Word/PDF file and submit it to the iLearn submission page.</u>

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

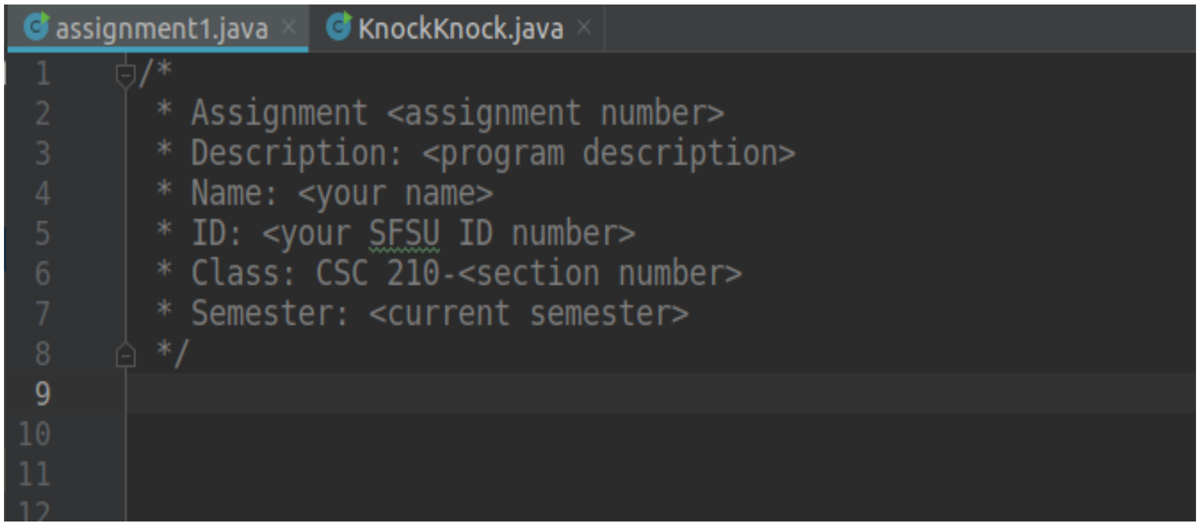
In this assignment, you are tasked with writing descriptive headers and comments.

You can write comments in two ways:

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

1. **Include a proper header at the top of every Java file.** 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: Example of the header that your program needs to have.

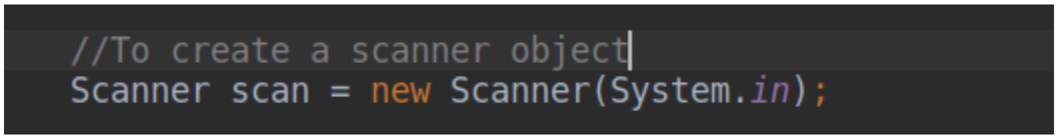


assignment1.java × KnockKnock.java ×

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

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

Figure 2: Example of writing single-line comment before each statement.



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