Assignment 4 Instructions

1. Assignment 01: **50 points total with 5 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: **9/28-2020 at 11:59 PM**

WHAT TO SUBMIT

Submit 4 files to iLearn by the deadline. [45pts + 5 E.C. pts = 50 points]

- 2 Java Files: Please submit 2 files to iLearn: **BmiMethods.java, Dice.java [30 points]**
- 1 PDF 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-4-report.pdf". Fill this out with screenshots then save it as Word or PDF
- 1 Reflection with a Buddy on iLearn
- 1 Extra Credit Post: Ask and Answer a question on iLearn

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**
4. **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>

*/
```

Assignment 4

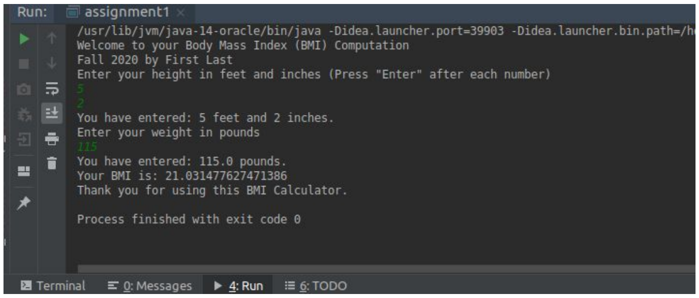
BODY MASS INDEX (BMI) COMPUTATION PRO

❐Part 1: BMI Using Methods [15 points]

1. Please do/redo the Part 2 of Assignment 03 using at least 2 methods.
    - Name our methods bmiStd and bmiPro.
    - bmiStd behaves like the standard version of our BMI.java program and bmiPro behaves like the TableBMI.java version

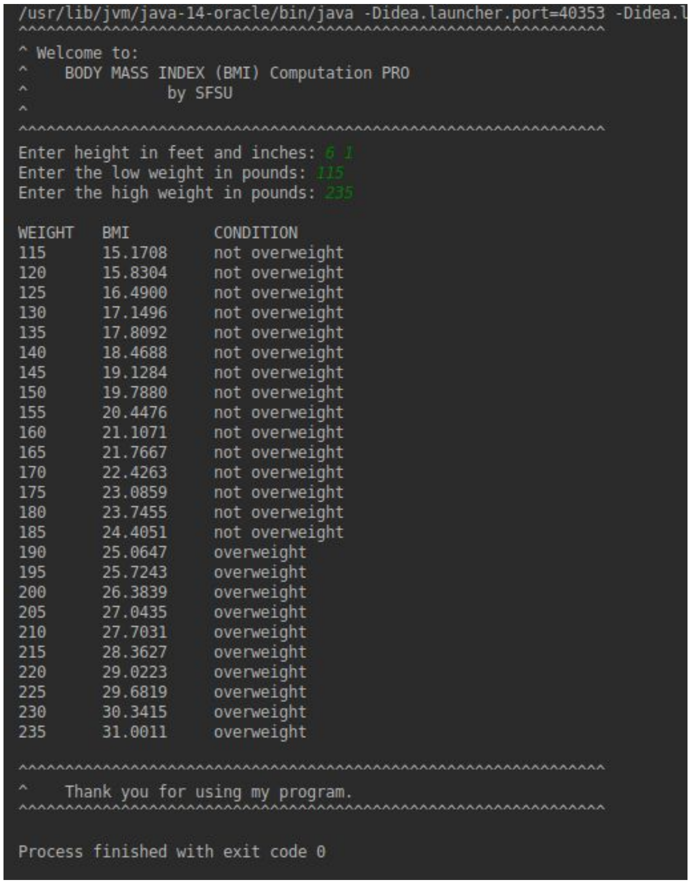
2. Our program must produce an identical output as that of Assignment 03 Part 1 for Pro version, and Assignment 01 for Standard version

OUTPUT OF SAMPLE RUN FROM ASSIGNMENT 1



```
/usr/lib/jvm/java-14-oracle/bin/java -Didea.launcher.port=39903 -Didea.launcher.bin.path=/h
Welcome to your Body Mass Index (BMI) Computation
Fall 2020 by First Last
Enter your height in feet and inches (Press "Enter" after each number)
5
2
You have entered: 5 feet and 2 inches.
Enter your weight in pounds
115
You have entered: 115.0 pounds.
Your BMI is: 21.031477627471386
Thank you for using this BMI Calculator.

Process finished with exit code 0
```

OUTPUT OF SAMPLE RUN FROM ASSIGNMENT 3 PART 1



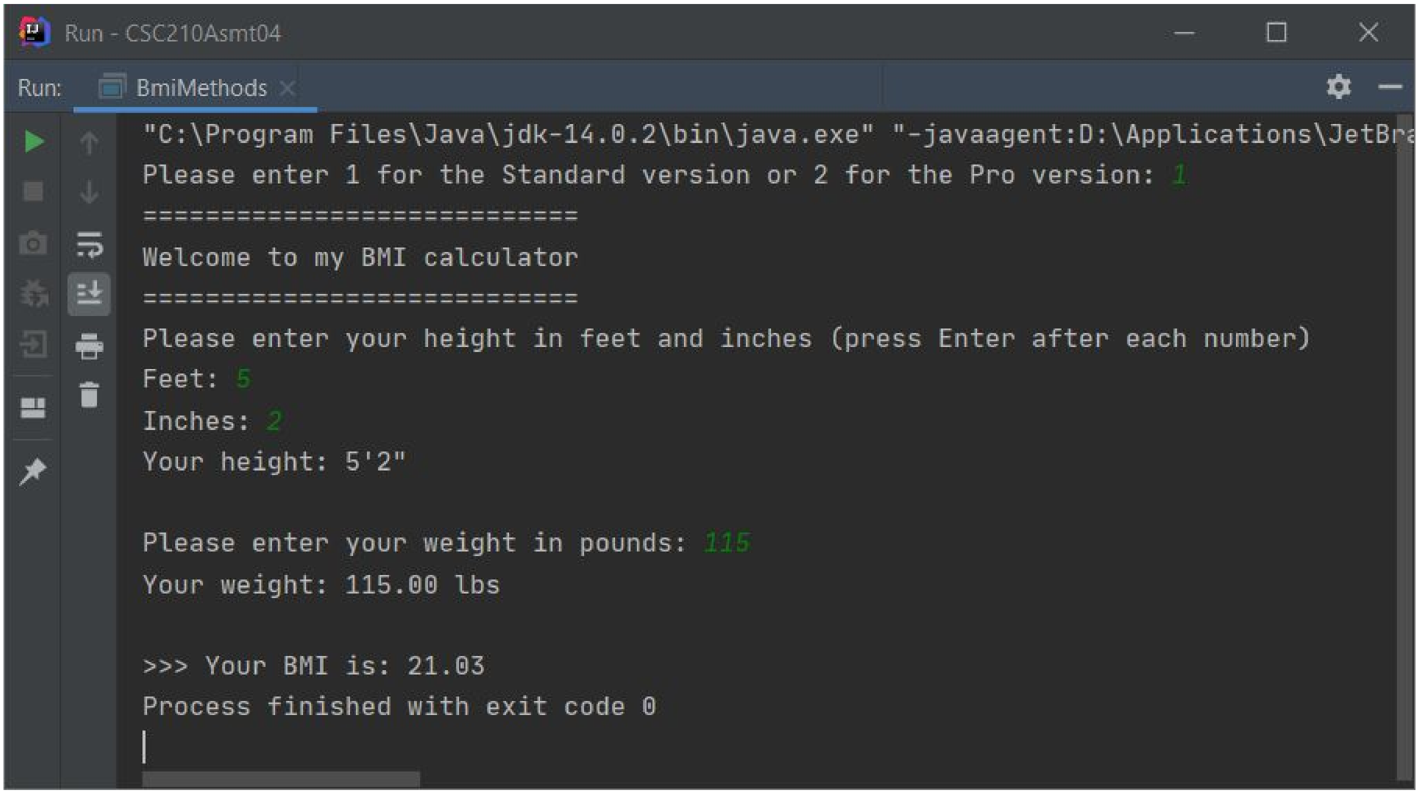
```
/usr/lib/jvm/java-14-oracle/bin/java -Didea.launcher.port=40353 -Didea.l
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

^ 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
```

SUBMISSION INSTRUCTIONS

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

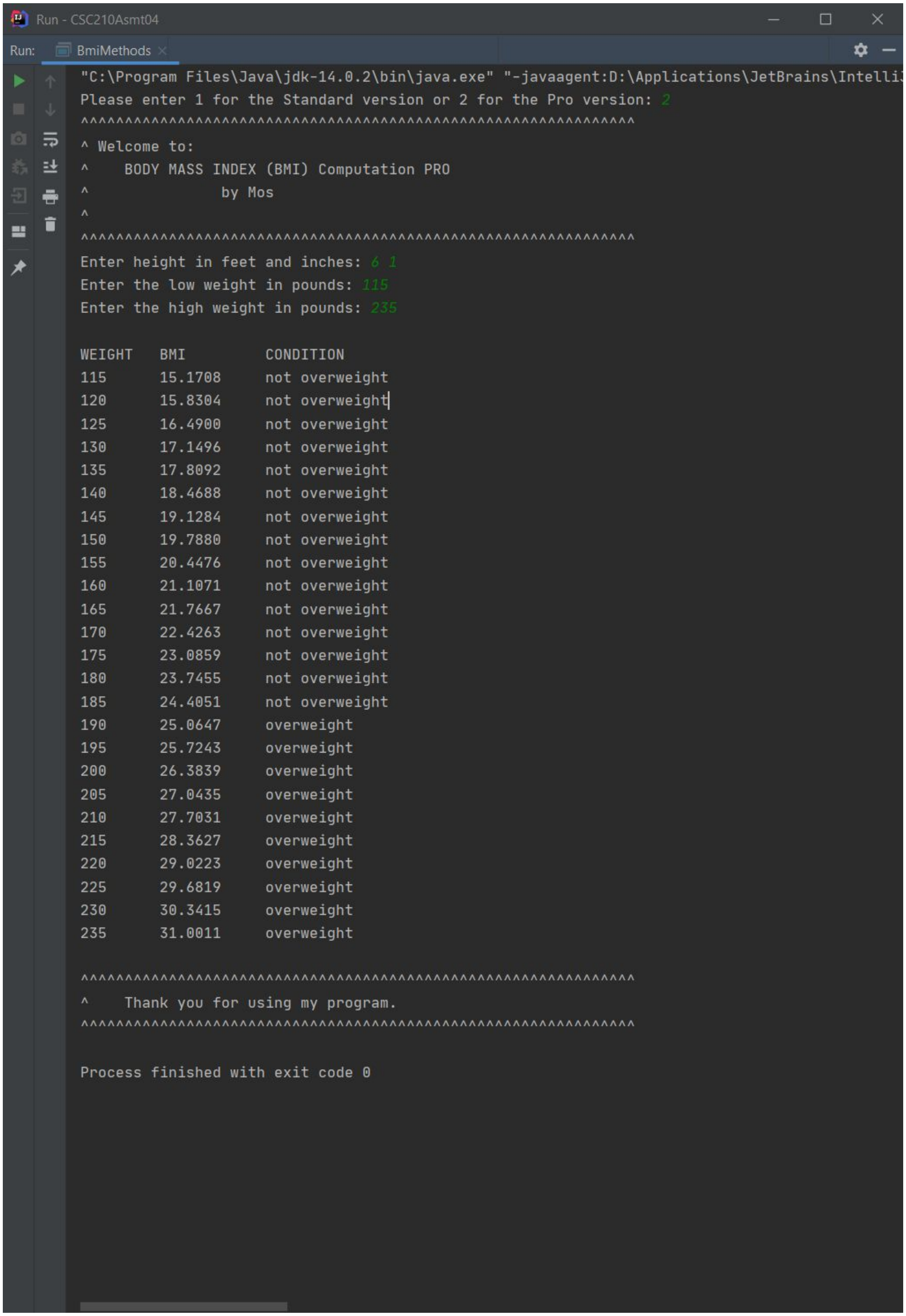
2. Take a screenshot of both outputs of your program and paste it here

Output of Standard Version :



```
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:D:\Applications\JetBra
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: 115
Your weight: 115.00 lbs

>>> Your BMI is: 21.03
Process finished with exit code 0
```

Output of Pro Version :



```
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:D:\Applications\JetBrains\IntelliJ
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: 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: Tracking the Value of a Dice [15 points]

Please write a Java program to track the count of each face of a dice as the user chooses to throw the dice several times.

You may use this Java file as a starting point, download and please edit this file to complete our program.

We are given 1 dice , with 6 face values : 1, 2, 3, 4, 5, and 6.
Your task will be to create a program that will

1. Prompt the user for a number of throws, anywhere between 1 and 1 million times.
    - You can assume the user will not enter anything more than 10_000_000 for this assignment
    - We don’t a negative value or a very high number
    - We want to use a Sentinel value. For my example I have used the integer 0
2. Unless the user enters a Sentinel value, of your choosing, you will continue to prompt the user for another number of throws
3. Outputs of Valid and Other outputs should look exactly like the below example

Example output: of 10000 throws

EAch of the output represent the count for when each face is selected

first loop/ throw

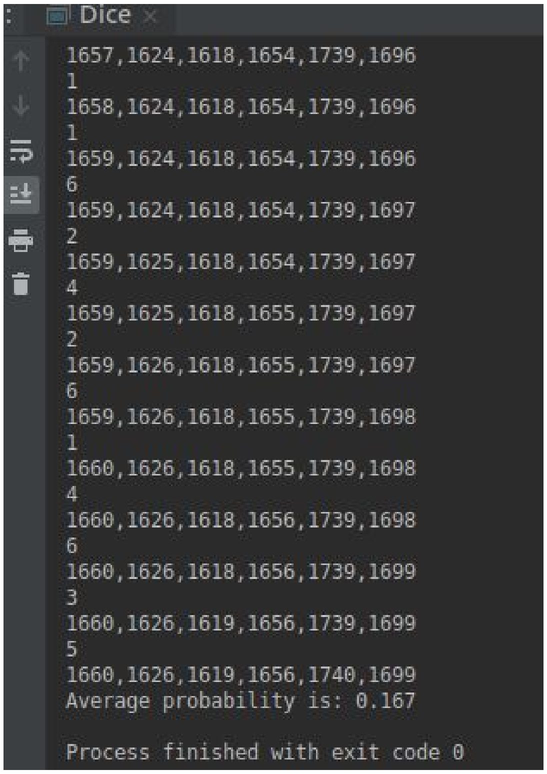
0,0,0,0,0,0 // dice face 1, 2, 3, 4, 5, 6

0,1,0,0,0,0//

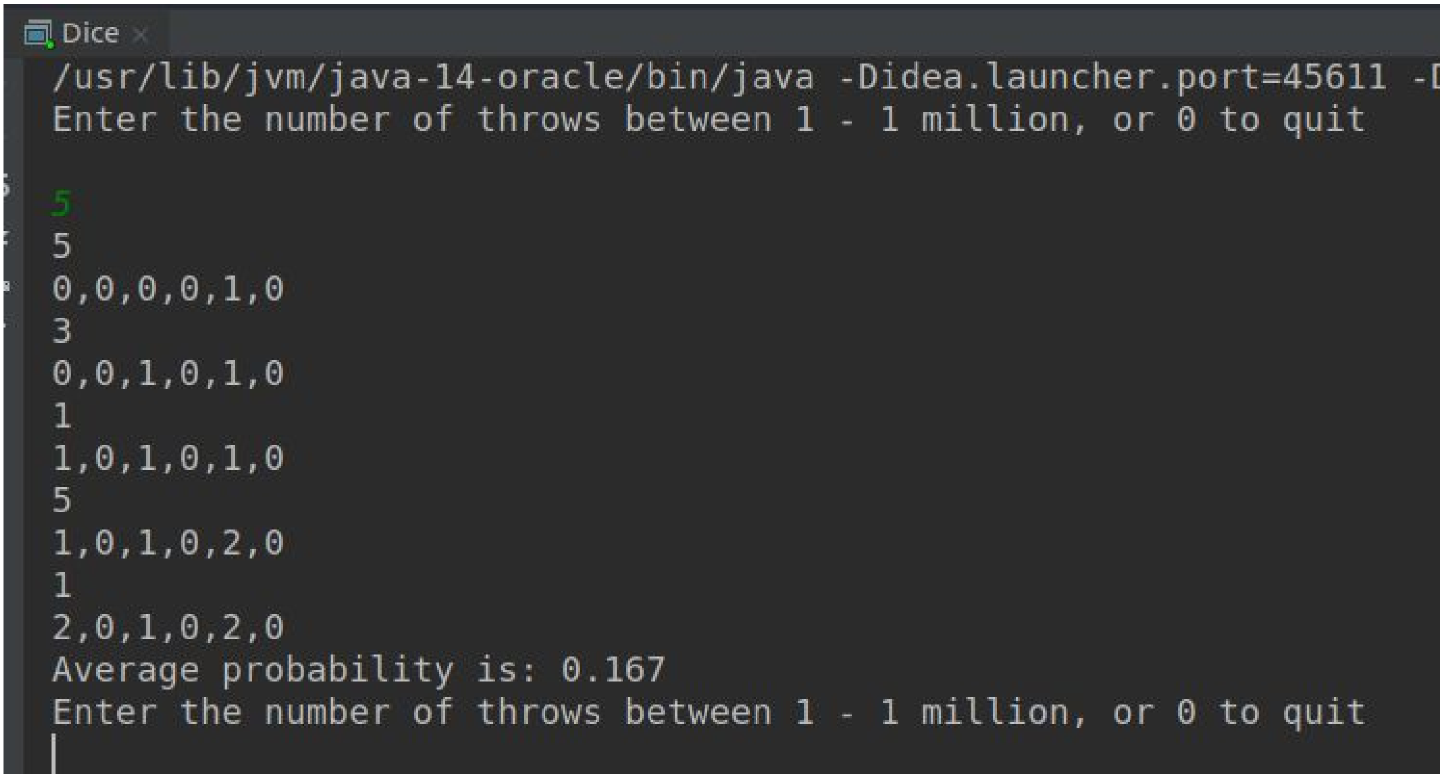
count1, count2, count3....

156, 135, 125 , 145, 100 = what is the average

Take that average/ divided by total throws ==1/6



```
1657,1624,1618,1654,1739,1696
1
1658,1624,1618,1654,1739,1696
1
1659,1624,1618,1654,1739,1696
6
1659,1624,1618,1654,1739,1697
2
1659,1625,1618,1654,1739,1697
4
1659,1625,1618,1655,1739,1697
2
1659,1626,1618,1655,1739,1697
6
1659,1626,1618,1655,1739,1698
1
1660,1626,1618,1655,1739,1698
4
1660,1626,1618,1656,1739,1698
6
1660,1626,1618,1656,1739,1699
3
1660,1626,1619,1656,1739,1699
5
1660,1626,1619,1656,1740,1699
Average probability is: 0.167

Process finished with exit code 0
```



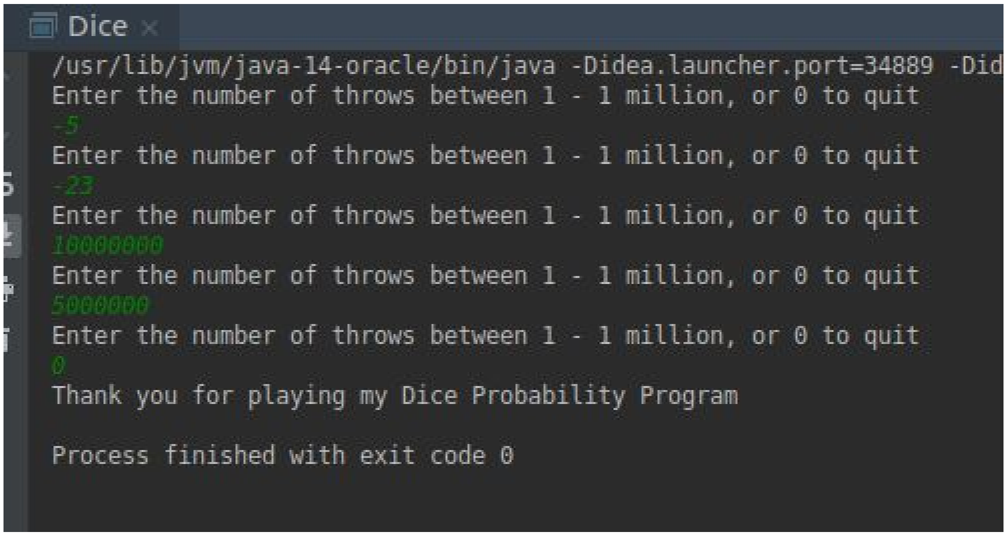
```
/usr/lib/jvm/java-14-oracle/bin/java -Didea.launcher.port=45611 -D
Enter the number of throws between 1 - 1 million, or 0 to quit

5
5
0,0,0,0,1,0
3
0,0,1,0,1,0
1
1,0,1,0,1,0
5
1,0,1,0,2,0
1
2,0,1,0,2,0
Average probability is: 0.167
Enter the number of throws between 1 - 1 million, or 0 to quit
```

Print out will be

1. Print each count ,

2. Print each dice face

Invalid value and Sentinel value output



```
/usr/lib/jvm/java-14-oracle/bin/java -Didea.launcher.port=34889 -Did
Enter the number of throws between 1 - 1 million, or 0 to quit
-5
Enter the number of throws between 1 - 1 million, or 0 to quit
-23
Enter the number of throws between 1 - 1 million, or 0 to quit
10000000
Enter the number of throws between 1 - 1 million, or 0 to quit
5000000
Enter the number of throws between 1 - 1 million, or 0 to quit
0
Thank you for playing my Dice Probability Program

Process finished with exit code 0
```

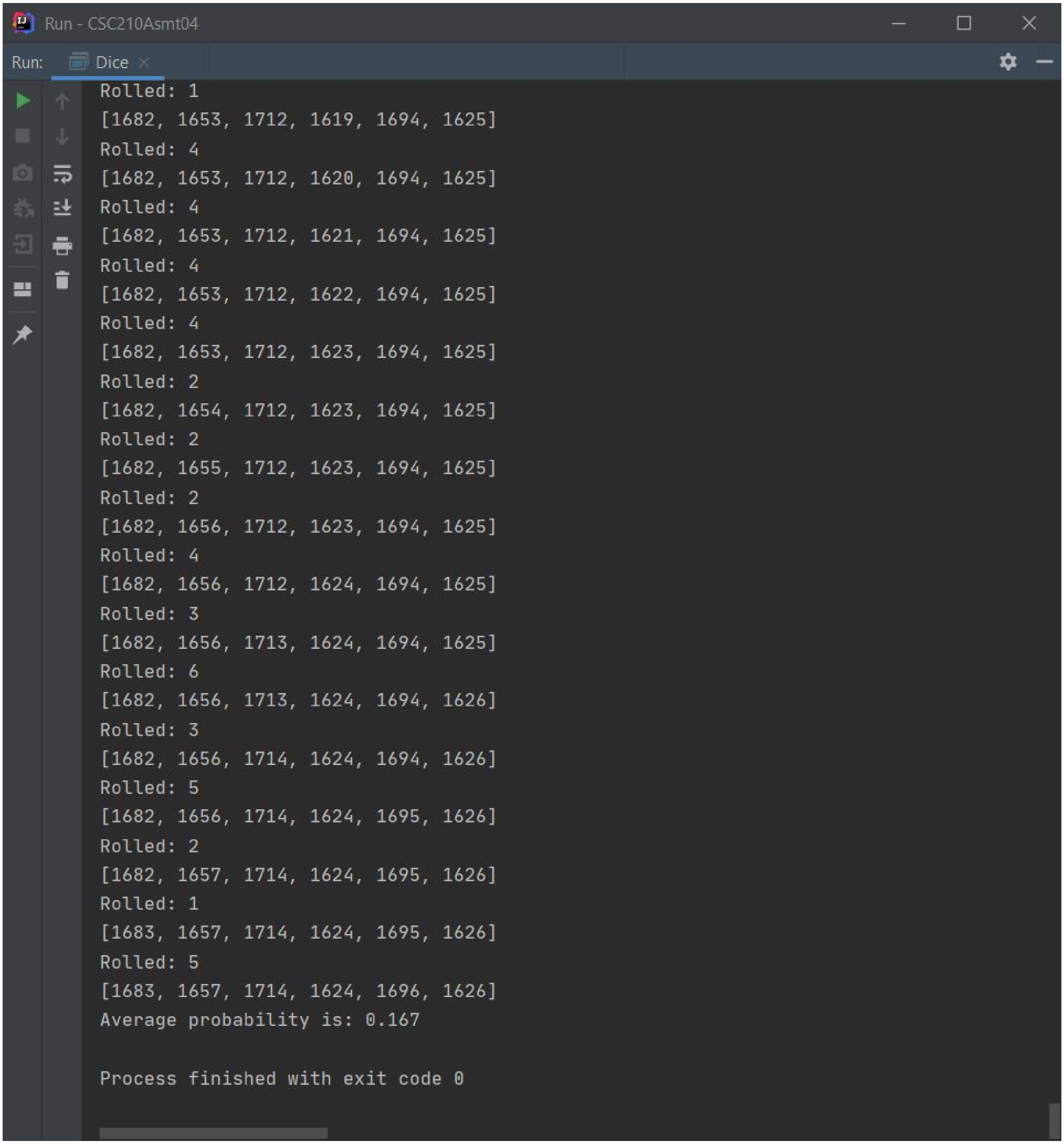
The Objective of this assignment is to
A. Learn how to build a fully functional program, which validates users, by using methods, loops, selections, and data types
B. To prove that, after a large amount of the dice throw, you do see the proof of the mathematical probability of each face of the dice as ⅙ for each face

SUBMISSION INSTRUCTIONS

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

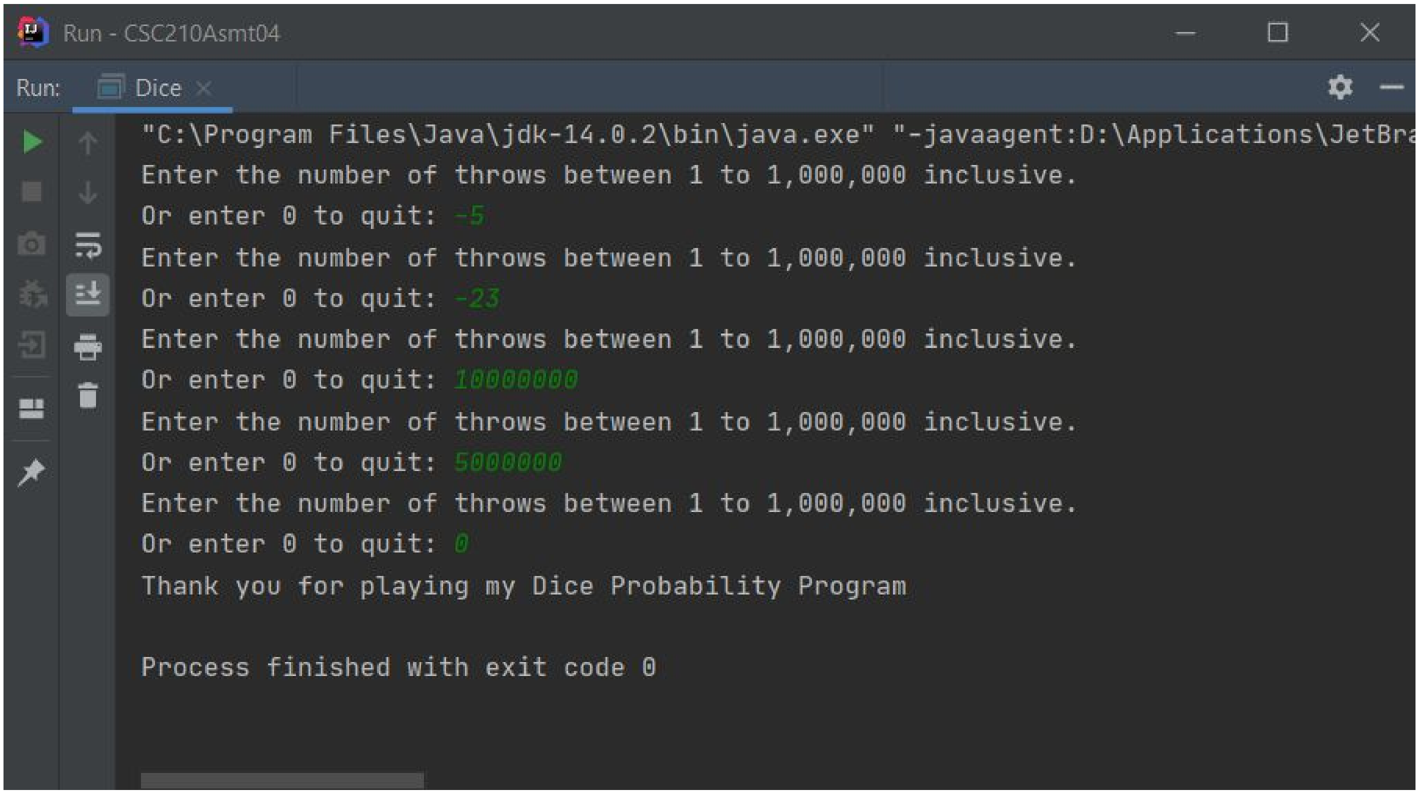
2. Take a screenshot of the output of your program showing the valid output and the average probability of each face as ⅙ and the output with sentinel and invalid values (negative and bigger than 1 million; If you test with too big a value you may run into errors, just test with 10 million or less)

Valid Output :



```
Rolled: 1
[1682, 1653, 1712, 1619, 1694, 1625]
Rolled: 4
[1682, 1653, 1712, 1620, 1694, 1625]
Rolled: 4
[1682, 1653, 1712, 1621, 1694, 1625]
Rolled: 4
[1682, 1653, 1712, 1622, 1694, 1625]
Rolled: 4
[1682, 1653, 1712, 1623, 1694, 1625]
Rolled: 2
[1682, 1654, 1712, 1623, 1694, 1625]
Rolled: 2
[1682, 1655, 1712, 1623, 1694, 1625]
Rolled: 2
[1682, 1656, 1712, 1623, 1694, 1625]
Rolled: 4
[1682, 1656, 1712, 1624, 1694, 1625]
Rolled: 3
[1682, 1656, 1713, 1624, 1694, 1625]
Rolled: 6
[1682, 1656, 1713, 1624, 1694, 1626]
Rolled: 3
[1682, 1656, 1714, 1624, 1694, 1626]
Rolled: 5
[1682, 1656, 1714, 1624, 1695, 1626]
Rolled: 2
[1682, 1657, 1714, 1624, 1695, 1626]
Rolled: 1
[1683, 1657, 1714, 1624, 1695, 1626]
Rolled: 5
[1683, 1657, 1714, 1624, 1696, 1626]
Average probability is: 0.167

Process finished with exit code 0
```

Invalid and Sentinel Output :



```
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:D:\Applications\JetBra
Enter the number of throws between 1 to 1,000,000 inclusive.
Or enter 0 to quit: -5
Enter the number of throws between 1 to 1,000,000 inclusive.
Or enter 0 to quit: -23
Enter the number of throws between 1 to 1,000,000 inclusive.
Or enter 0 to quit: 10000000
Enter the number of throws between 1 to 1,000,000 inclusive.
Or enter 0 to quit: 5000000
Enter the number of throws between 1 to 1,000,000 inclusive.
Or enter 0 to quit: 0
Thank you for playing my Dice Probability Program

Process finished with exit code 0
```

❐Part 3: Reflect with a Buddy 50 words [15 points]

1. Find your buddy. Look at the Study Buddy sheet and pair up with your assigned buddy. Or find someone to partner this task with.

2. You can set up a Slack, Discord, or a Zoom with your buddy to asynchronously or synchronously talk about anything, and also reflect on what was helpful and not helpful in completing this homework.

3. Each student will need to create 1 post to mark this assignment as complete.

Have fun with this one!!

👋 Assignment 4 Part 3: Reflection with a Buddy

The following reflection is identical to the one posted on iLearn.

This assignment was pretty fun and interesting. I thought that the explanations in the assignments were clear for the most part. Although, at times the examples in the assignment can be unclear on what it is demonstrating. However, this was quickly resolved later on via Slack and also during the class. My study buddy is Amber Hartigan. I found out that Amber was new to computer science and programming so I try to help her by explaining things in plain language and through other relatable real-life examples. In this particular assignment, I helped Amber with the syntaxes for defining and calling methods.