

**CEGEP VANIER COLLEGE**  
**Department of Computer Science**

**420-101-VA PROGRAMMING 1 sections 1 &2**

**Assignment\_01:** Due October 17<sup>th</sup>, 2023, at 11:59pm

**Prof. Nagat Drawel**

---

**Overview:**

The purpose of this assignment is to learn:

1. For, While, Do While looping Statements
2. Break and Continue statements
3. How to generate a random integer number in a specific range

**Notes:**

1. Please make sure to follow the programming standards (e.g., naming conventions, comments, @author); Keep your code clean (Indentation, Spaces, etc.,) not doing so will result in loss of marks.
2. Please demonstrate your working programs to your teacher.
3. Before turning in your program, make sure you have followed all the instructions. Always test, and retest that your program compiles and runs successfully before submitting it.
4. Please submit one .java file for each task (4 .java files in total, Omnivox accepts multi-submission). **DO NOT ZIP YOUR FILES.**

**Task 01**

Convert the pseudocode below into Java code and place it in the main method after the variable declarations. You will be using several control structures: a while loop and an if-else-if statement nested inside another if statement. Use the indenting of the algorithm below to help you decide what is included in the loop, what is included in the if statement, and what is included in the nested if-else-if statement. To “roll” the dice, use the random method (**Math.random()**) to generate an integer from 1 to 6

*Repeat while the number of dice rolls are less than the number of times the dice should be rolled.*

*Get the value of the first die by “rolling” the first die*

*Get the value of the second die by “rolling” the second die*

*If the value of the first die is the same as the value of the second die*

*If the value of first die is 1*

*Increment the number of times snake eyes were rolled*

*Else if value of the first die is 2*

*Increment the number of times twos were rolled*

*Else if value of the first die is 3*

*Increment the number of times threes were rolled*

*Else if value of the first die is 4*

*Increment the number of times fours were rolled*  
*Else if value of the first die is 5*  
*Increment the number of times fives were rolled*  
*Else if value of the first die is 6*  
*Increment the number of times sixes were rolled*  
*Increment the number of times the dice were rolled*

Run your code several times. You should get different results than the first time  
*First run:*

```
You rolled snake eyes 309 out of 10000 rolls.  
You rolled double twos 264 out of 10000 rolls.  
You rolled double threes 289 out of 10000 rolls.  
You rolled double fours 310 out of 10000 rolls.  
You rolled double fives 267 out of 10000 rolls.  
You rolled double sixes 274 out of 10000 rolls.  
BUILD SUCCESSFUL (total time: 0 seconds)
```

*Second run:*

```
You rolled snake eyes 274 out of 10000 rolls.  
You rolled double twos 261 out of 10000 rolls.  
You rolled double threes 274 out of 10000 rolls.  
You rolled double fours 297 out of 10000 rolls.  
You rolled double fives 293 out of 10000 rolls.  
You rolled double sixes 271 out of 10000 rolls.  
BUILD SUCCESSFUL (total time: 0 seconds)
```

## **Task 02**

A large company pays its salespeople on a commission basis. The salespeople receive \$200 per week plus 9% of their gross sales for that week. For example, a salesperson who sells \$5,000 worth of merchandise in a week receives \$200 plus 9% of \$5,000, or a total of \$650. You've been supplied with a list of the items sold by each salesperson. The values of these items are shown below:

Item	Value
1	239.99
2	129.75
3	99.95
4	350.89

Write a Java program that inputs one salesperson's items sold for last week and calculates and displays that salesperson's earnings. There's no limit to the number of items that can be sold.

### Sample Output:

```
Enter number sold of product #1: 12
Enter number sold of product #2: 10
Enter number sold of product #3: 4
Enter number sold of product #4: 7
Earnings this week: $833.01
BUILD SUCCESSFUL (total time: 28 seconds)
```

### Task 3:

Write a Java program that asks the user to enter an average of five courses. The program should display a bar chart comparing each course average. Create each bar in the bar chart by displaying a row of asterisks. Each asterisk should represent 10% of the averages. Here is an example of the program's output

### Sample Output:

```
-----
Enter the average of Programming I:  89
Enter the average of Database: 80
Enter the average of Computer Environment: 67
Enter the average of Networks: 99
Enter the average of Programming II: 70

AVERAGE BAR CHART
*****
*****
*****
*****
*****
BUILD SUCCESSFUL (total time: 33 seconds)
```

### Task 4

Create a program that generates a multiplication table comprising 25 rows of calculations. Allow the user to input the first and last base values for the multiplication table. The table should have columns starting with the first base input and ending with the last base input. Each row in the table corresponds to a different multiplier, ranging from 1 to 25. For each multiplier, calculate the product with each base value within the specified range. Ensure that the program validates the base values entered by the user, allowing values between 2 and 8. Display an error message if an invalid base is entered. Finally, format the multiplication table in an aesthetically pleasing manner for easy readability.

An example of output produced when 2 and 8 are entered appears below:

Enter the first base value (between 2 and 8): 2

Enter the last base value (between 2 and 8): 8

Multiplication Table:

	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8
2	4	6	8	10	12	14	16
3	6	9	12	15	18	21	24
4	8	12	16	20	24	28	32
5	10	15	20	25	30	35	40
6	12	18	24	30	36	42	48
7	14	21	28	35	42	49	56
8	16	24	32	40	48	56	64
9	18	27	36	45	54	63	72
10	20	30	40	50	60	70	80
11	22	33	44	55	66	77	88
12	24	36	48	60	72	84	96
13	26	39	52	65	78	91	104
14	28	42	56	70	84	98	112
15	30	45	60	75	90	105	120
16	32	48	64	80	96	112	128
17	34	51	68	85	102	119	136
18	36	54	72	90	108	126	144
19	38	57	76	95	114	133	152
20	40	60	80	100	120	140	160
21	42	63	84	105	126	147	168
22	44	66	88	110	132	154	176
23	46	69	92	115	138	161	184
24	48	72	96	120	144	168	192
25	50	75	100	125	150	175	200