

# AP Computer Science Java

## Mr. Clausen

### Program 4A

**Program 4A** LastNameFirstNameP4A (Die Roll: 25 points)

Write a program that asks the user to enter a number between 1 and 1000. Use a primed while loop to “error trap” and make sure that the number that the user enters is between these limits. Use constants for the LOWER\_LIMIT and for the UPPER\_LIMIT of 1 and 1000 respectively. Use a “for loop” that counts from 1 to the user’s number and generates a random integer between 1 and 6 to simulate the rolling of a die as many times as the user asked. Display these random numbers in a table format displaying 10 numbers across each row each separated from the other by 2 spaces (this will require an “if statement” with the modulus operator). You will also need to keep track of how many times a 1 was rolled, a 2 was rolled, etc. **DO NOT USE a break statement to escape from any type of loop in this class regardless of what the book says or models.**

- 1) Type comments at the beginning of the program to display your name and other information just like those used for program 2A.
- 2) Import the packages:  
import java.util.Scanner;  
import java.util.Random;
- 3) Declare the class name in the format LastNameFirstNameP4A. Don’t forget that the filename needs to be the same when you save your program. This would be a good time to save your program if you haven’t done so already, LastNameFirstNameP4A.java.
- 4) Declare the main method:  
public static void main(String [] args)
- 5) Next declare all of the constants necessary for this program. Declare a constant of type **int** for LOWER\_LIMIT = 1, and declare a constant of type **int** named UPPER\_LIMIT = 1000. These constants will be used to error trap the user’s input for how many times we will simulate rolling a die.
- 6) Instantiate an object of the Scanner class in order to use input from the keyboard. Also instantiate an object of the Random class in order to use random numbers. Use the statements:  
Scanner reader = new Scanner(System.in);  
Random generator = new Random( );
- 7) Declare all of the variables necessary for this program. You will need integers to store the number of times to roll a die, the current value of the die roll, the number of times a

one, two, three, four, five, and/or six are rolled. Make sure that you use descriptive variable names to create readable, self-documenting code.

- 8) Type the following comment:

```
//-----Display My Information-----
```

Follow this comment with println statements to display your name and period output just like those used for program 2A,

- 9) For the Input section, type the following comment:

```
//-----Input-----
```

Ask the user to enter the upper limit for the number of times that they would like to roll the die. Make your program user friendly by using “print” (without the ln) statements prompting the user for this. Use a “primed while loop” to “error trap” this value to make sure that it is between LOWER\_LIMIT and UPPER\_LIMIT inclusive.

- 10) For the Calculations section, type the following comment:

```
//-----Calculations & Output-----
```

Use a “for” loop to repeat rolling a die from one to the number of times that the user entered to roll a die and generates a random integer between 1 and 6 to simulate the rolling of a die. Declare the loop counter inside the loop header; call this variable “counter”. Display these random numbers in a table format displaying 10 numbers across each row each separated from the other by 2 spaces (this will require an “if statement” with the modulus operator). You will also need to keep track of how many times a 1 was rolled, a 2 was rolled, etc. If you are new to programming, you can use sequential selection statements (“if statements”) just for this program only to keep track of how many times each number was rolled. If you are an experienced programmer, use nested (extended) “if else statements” (even though we won’t cover this until Chapter 6).

- 11) For the Output section, type the following comment:

```
//-----Output-----
```

For the output portion of your program, display how many times each number was rolled. Echo back the upper limit to remind us how many rolls of the die we chose. Make this output easy to read and formatted nicely.

Use blank lines to separate each of the program sections listed in all of the steps above. When you are finished with your program, have tested it thoroughly to make sure that your calculations are correct, and are sure that you don’t need to make any changes, then save your program in the “W” network mapping, in the Program 4A folder. **Turn in the file named LastNameFirstNameP4A.java, I don’t need the “.class” file for this program.**