**Balam Indira Priyadarsini**

**Bridge course Assignment-Day 3**

**Session 1:**

**Countdown**

**Problem Statement:** Print numbers from 10 to 1,then “Blastoff!”

**Algorithm:**

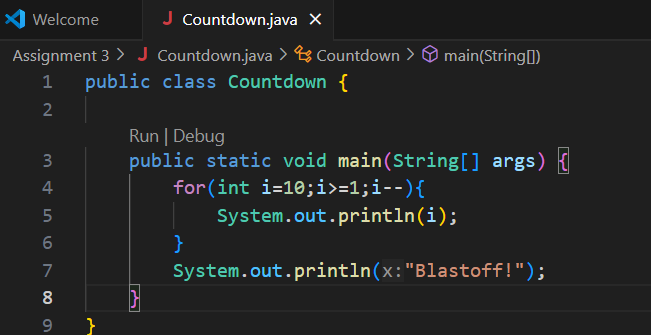
1. Start from number 10.

2. Loop down to 1, decrementing by 1 each time.

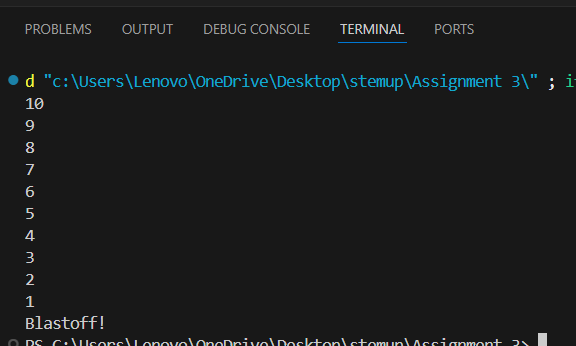
3. Print each number in the loop.

4. After the loop ends, print “Blastoff!”

**Pseudo Code:**



**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| **Test Case no** | **Input range** | **output** |
| Tc1 | 10 to1 | 10 9 8 7 6 5 4 3 2 1 Blastoff! |
| Tc2 | 5 to 1 | 5 4 3 2 1 Blastoff! |
| Tc3 | 1 to 0 | 1 Blastoff! |

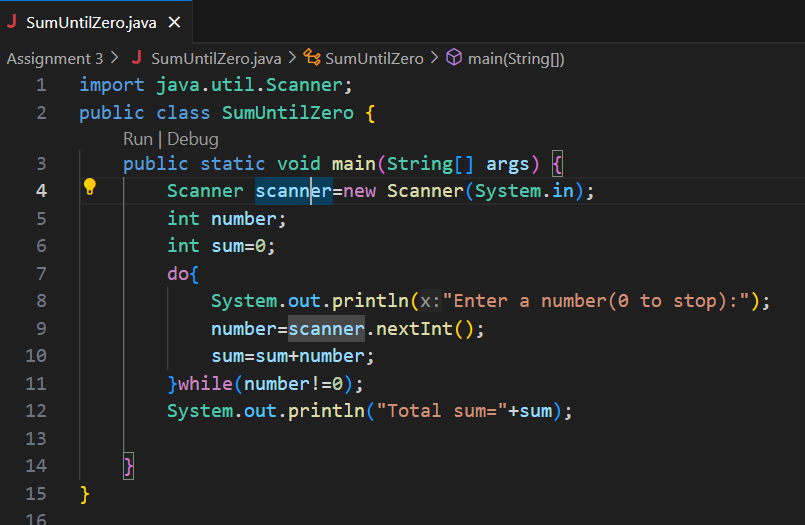
**Sum Until Zero**

**Problem Statement:** Ask user for numbers repeatedly until they enter 0.Sum and print the total.

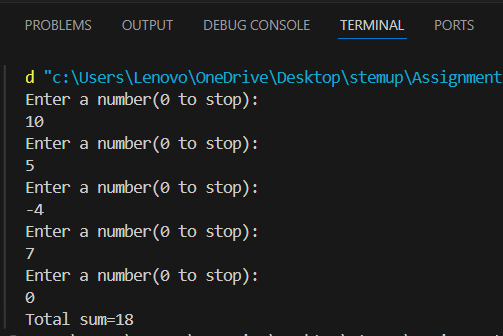
**Algorithm:**

1. start the program
2. Ask user to enter number
3. Add the number to sum
4. Stop when the number is equal to 0
5. Print the sum value

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| **Test Cases no** | **Input values** | **Output** |
| **Tc1** | **10,5,-4,7,0** | **Total sum=18** |
| **Tc2** | **0** | **Total sum=0** |

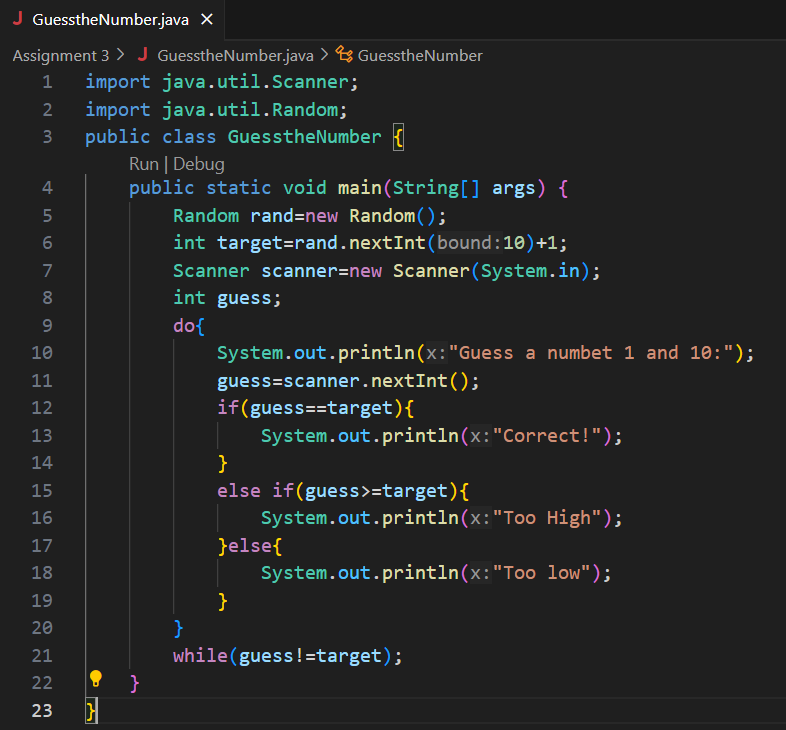
**Guess the Number**

**Problem Statement:**Generate a random number between 1 and 10.Ask user to guess.Provide feedback and loop until correct.

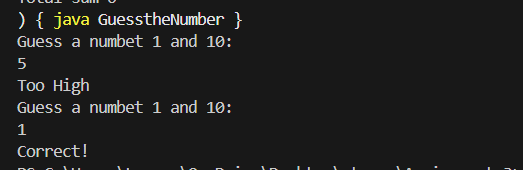
**Algorithm:**

1. Generate a random number between 1 and 10 and store it in a variable
2. It will ask to enter a guess number
3. Convert the input to an integer
4. If guess is equal to secret number it print correct!
5. If not it would show too high are too low upto you guess a correct number
6. Then exit from the loop.

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case no** | **Random number** | **User guesses** | **Expected output** |
| 1 | 1 | 5,1 | Too High,correct |
| 2 | 7 | 4,8,6,7 | Too low,Too high,Too low,correct |
| 3 | 4 | 7,4 | Too High,correct |

**Infinite Loop Debugging**

**Problem Statement:** Analyze and fix:

Int counter=0;

While (counter<5){

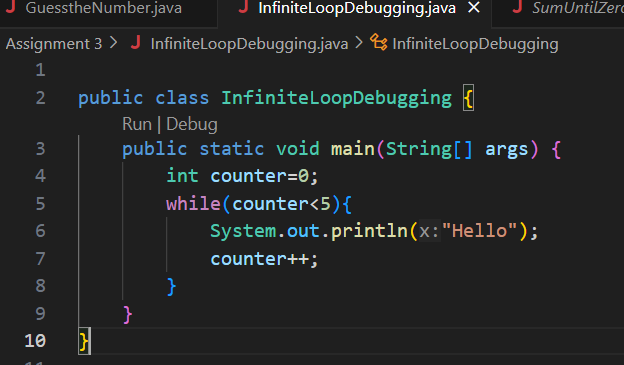
System.out.println(“Hello”);

}

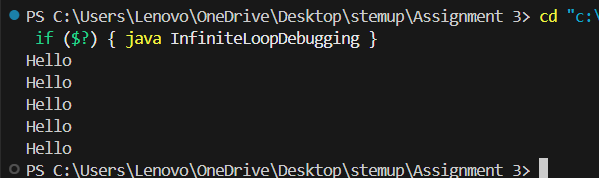
**Algorithm:**

1. Start a program
2. Check the initial counter to
3. While counter<5 ,Print “Hello”
4. And increment the counter by 1
5. Stop the counter when it reaches 5

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| **Test case no** | Initial counter | Output |
| 1 | 0 | “Hello” print 5 times |
| 2 | 3 | “Hello” print 2 times |
| 3 | -2 | “Hello” print 7 times |

**Section 2:**

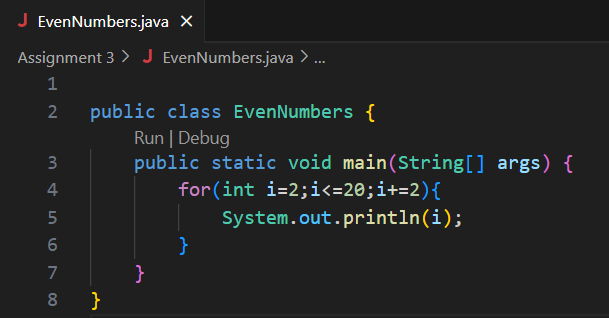
**Even Numbers:**

**Problem Statement:**­print even numbers from 2 to 20 using a for loop.

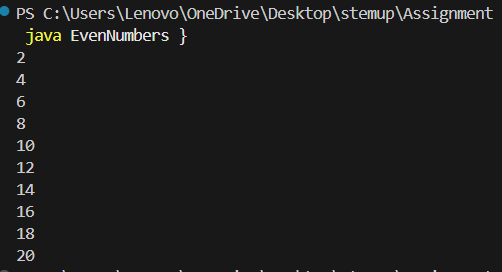
**Algorithm:**

1. Start a program with loop variable from 2.
2. Continue the loop upto the value less than equal to 20.
3. Increment the variable by 2 to get even number
4. Print the variables
5. End the loop when the condition fails

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| **Test case no** | Range | Output |
| 1 | 2 to 20 | 2 4 6 8 10 12 14 16 18 20 |
| 2 | 10 to 20 | 10 12 14 16 18 20 |
| 3 | -6 to 8 | -6 -4 -2 0 2 4 6 8 |

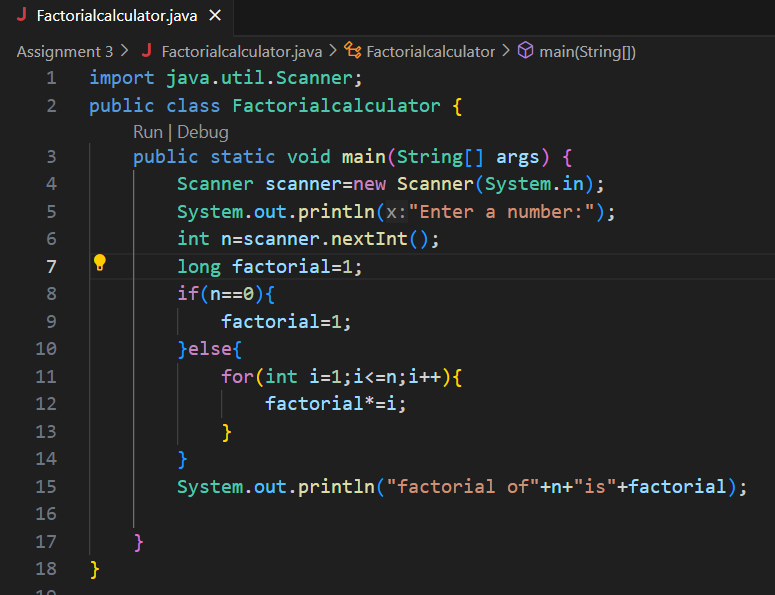
**Factorial calculator**

**Problem Statement:** Calculate n! for user input n. Handle edge case when n==0.

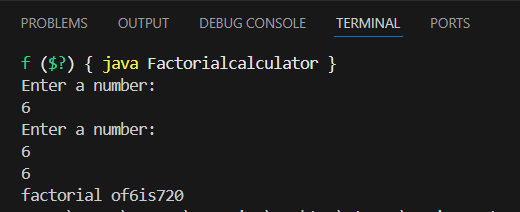
**Algorithm:**

1. Start a program by enter a number
2. If n==0,it should return as 1
3. Initialize the variable fact to 1
4. Used for loop from 1 to n values
5. This will multiply fact by loop variable
6. After loop display fact as the result

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| Test case | Input | Output |
| 1 | 6 | Factorial of 6 is 720 |
| 2 | 5 | Factorial of 5 is 120 |
| 3 | 10 | Factorial of 10 is 3628800 |

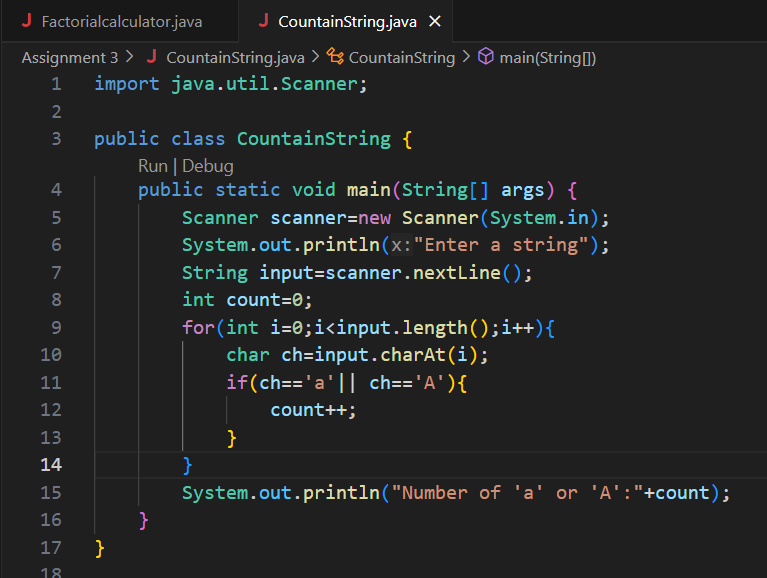
**Count ‘a’ in String**

**Problem Statement:** Ask for a string input. Count how many times ‘a’ or ‘A’ appears

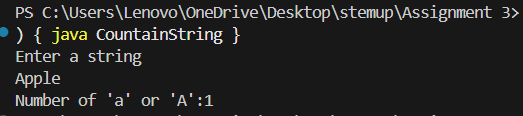
**Algorithm:**

1. Enter a string
2. Initialize a counter to 0.
3. If a character is ‘a’ or ‘A’, increment the counter.
4. After the loop, print the count

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| Test case | Input string | Output |
| 1 | “Apple” | 1 |
| 2 | “Indira” | 1 |
| 3 | “Hello World” | 0 |

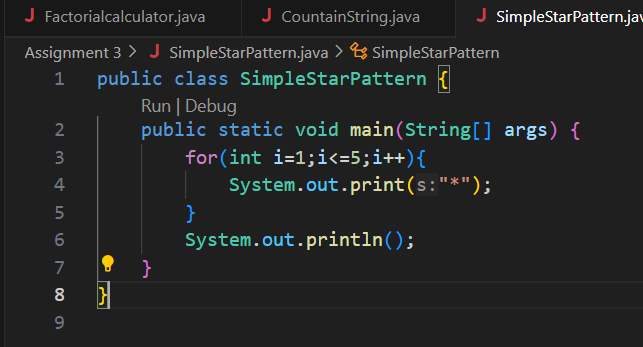
**Simple Star Pattern**

**Problem Statement:** Print: \*\*\*\*\* using one for loop

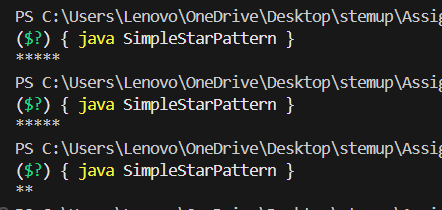
**Algorithm:**

1. 1.Use for loop and run for 5 time
2. For each iteration it will print the “\*”
3. After the loop end it exits and stop

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| Test case | Input | Output |
| 1 | 5 | \*\*\*\*\* |
| 2 | 2 | \*\* |
| 3 | 10 | \*\*\*\*\*\*\*\*\*\* |

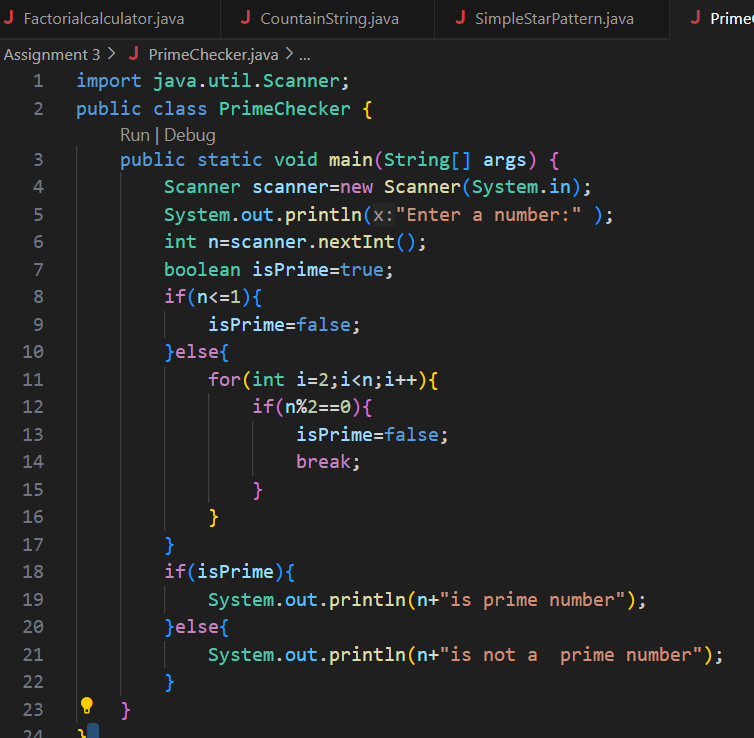
**Prime Checker:**

**Problem Statement:** Check if a number is prime using a loop and break.

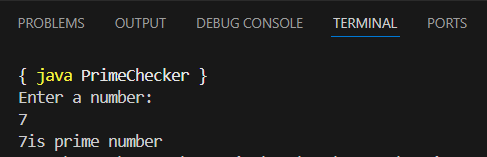
**Algorithm:**

1. Enter a number as n
2. If n is less than or equal to 1, is not a prim
3. Repeat for i from 2 to n-1
4. If n % i ==0,then it’s not prime and stop the loop
5. If the loop completes without finding a divisor, then n is prime.
6. Print whether a number is prime or not prime

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| Test case | Input | Output |
| 1 | 7 | Prime |
| 2 | 5 | Prime |
| 3 | 10 | Not Prime |

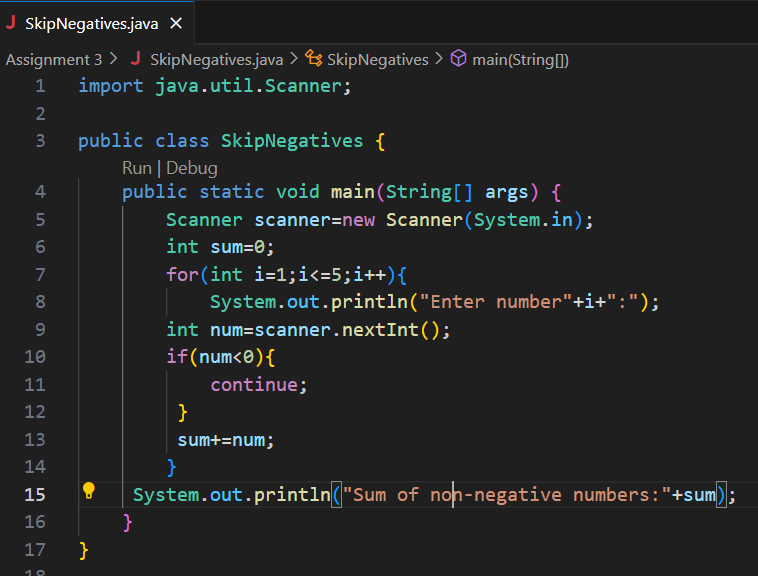
**Skip Negatives:**

**Problem Statement:** Input 5 numbers. Use continue to skip negative ones and sum the rest

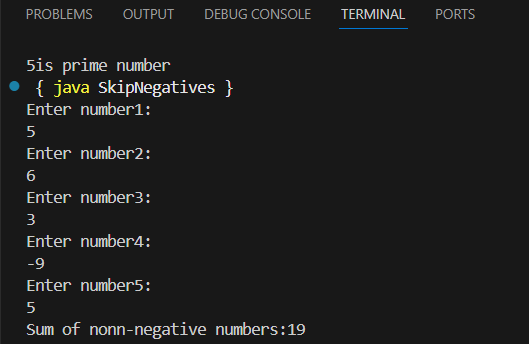
**Algorithm:**

1. Initialize sum=0 and counter for 5 inputs
2. Take a number as a input
3. If number is negative ,use continue to skip the rest of the loop
4. Add the number to sum
5. After loop print the final sum

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| Test case | Input | Output |
| 1 | 5,6,3,-9,5 | Sum of non-negative numbers:19 |
| 2 | 2, 3, 4, 5, 6 | Sum of non-negative numbers:20 |
| 3 | |  | | --- | |  |  |  | | --- | | 1, -2, 3, -4, 5 | | Sum of non-negative numbers:9 |

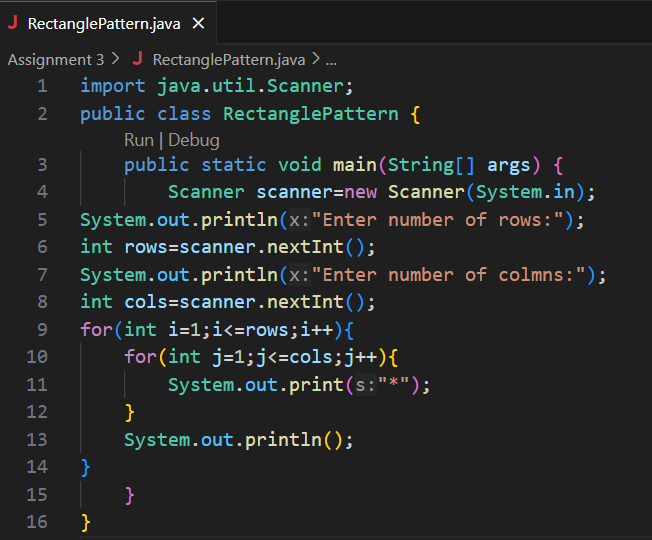
**Rectangle Pattern:**

**Problem Statement:** Input rows and columns ,print a rectangle of \*.

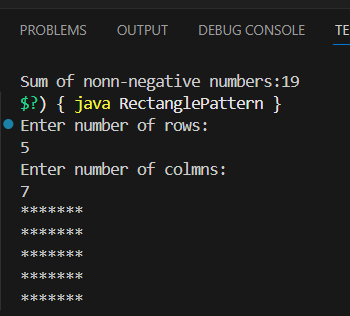
**Algorithm:**

1. Enter number of rows
2. Enter number of columns
3. Using for loop through each row and column print \* with out new line
4. Print one full row and move to next line

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| Test case | Input | Output |
| 1 | 5\*7 | 5 rows of 7 stars |
| 2 | 3\*5 | 3 rows of 5 stars |
| 3 | 10\*6 | 10 rows of 6 stars |

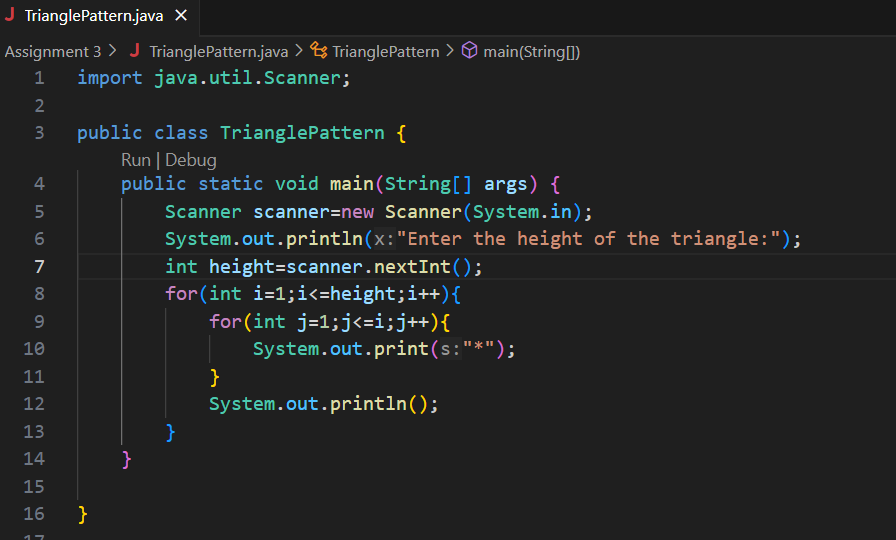
**Triangle Pattern:**

**Problem Statement:** Input height. Print right-angled triangle with \*.

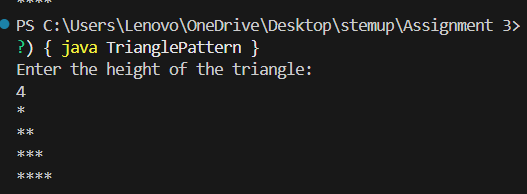
**Algorithm:**

1. Enter the height of the triangle
2. Loop from 1 to height by the iteration(row i)print i stars using another loop
3. After printing stars in each rows move to next line
4. Repeat it until the triangle complete

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| Test case | Input height | Output |
| 1 | 0 |  |
| 2 | 3 |  |
| 3 | 10 |  |

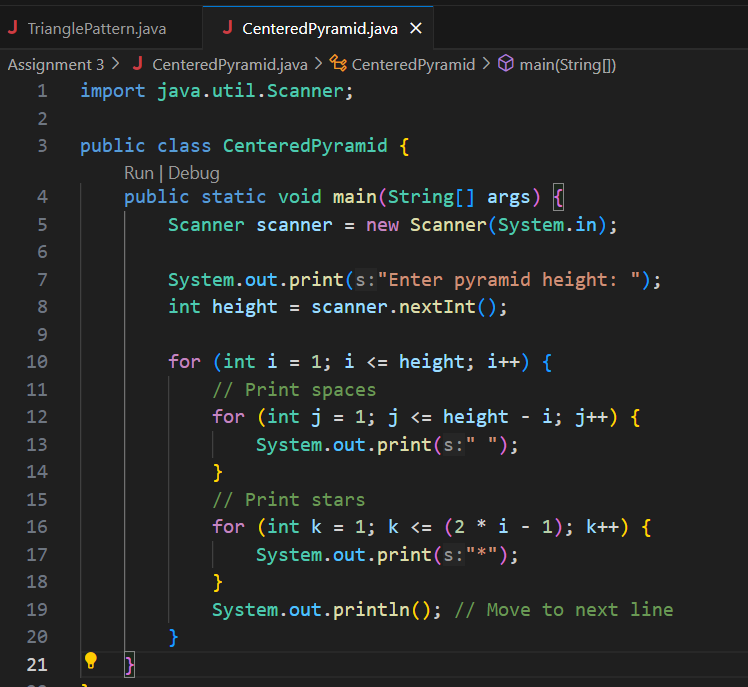
**Pyramid Pattern Challenge:**

**Problem Statement:**Input height.print centered pyramid.

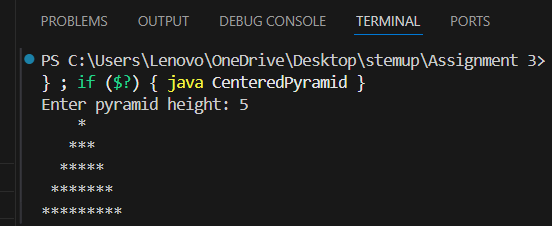
**Algorithm:**

1. Ask the user for the height of the pyramid.
2. Loop from 1 to height (row-wise):
3. Print (height - i) spaces.
4. Print (2 \* i - 1) stars.
5. Move to the next line.
6. End the loop.

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| Test case | Input | Output |
| 1 | 6 |  |
| 2 | 5 |  |
| 3 | 10 |  |

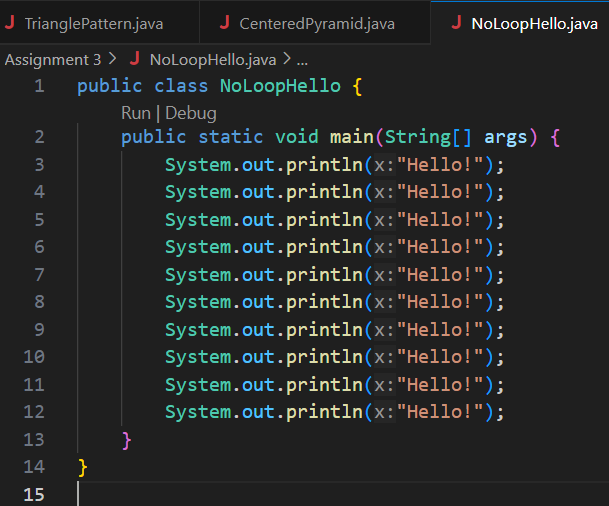
**Code Duplication:**

**Problem Statement:** Write how you would print “Hello!” 10 times without loops.Reflect on how loops make this easier for 1000 times.

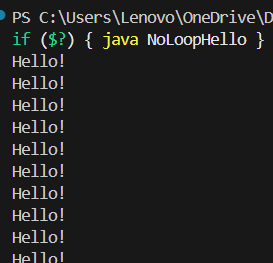
**Algorithm:**

1. Write the System.out.println("Hello!"); statement 10 times manually.
2. End program.

**Pseudo Code:**

****

**Output:**

****

**Testcases:**

|  |  |  |
| --- | --- | --- |
| Test case | Input | Output |
| 1 | 10 | 10 lines of "Hello! |
| 2 | 5 | 5 lines of "Hello! |