

Bridge Course – Day 3

Submitted By: Adnan Sameer

Activity 1: Repetitive Tasks List three tasks you perform regularly that involve repetition. For each:

1. What is being repeated?
2. What determines when it stops?

1.1 Algorithm:

1.1.1 Algorithm to solve coding problems daily:

Step 1: Start
Step 2: Input the number of problems to be solved (totalProblems)
Step 3: Repeat the following steps from i = 1 to totalProblems
Step 4: Display "Solving problem i..."
Step 5: After the loop ends, display "You have completed all problems."
Step 6: Stop

1.1.2 Algorithm to check emails:

Step 1: Start
Step 2: Set number of unread emails (e.g., 5)
Step 3: While unread emails > 0, repeat:
Step 4: Read one email
Step 5: Reduce unread email count by 1
Step 6: End loop when unread emails = 0
Step 7: Print "Inbox is clear!"
Step 8: Stop

1.1.3 Algorithm to attend class:

Step 1: Start
Step 2: Set total number of classes (e.g., 3)
Step 3: For each class from 1 to total classes, repeat:
Step 4: Print "Attending class [i]"
Step 5: After loop ends, print "All classes attended"
Step 6: Stop

1.2 Pseudocode:

1.2.1 Pseudocode for solving coding problems:

```
Start
Read totalProblems
For i ← 1 to totalProblems do
    Print "Solving problem i..."
End For
Print "You have completed totalProblems problems. Good job!"
Stop
```

1.2.2 Pseudocode for Checking mails:

```
Start
Set unreadEmails ← 5

While unreadEmails > 0 do
    Print "Reading an email..."
    unreadEmails ← unreadEmails - 1
End While

Print "Inbox is clear!"
Stop
```

1.2.3 Pseudocode for attending class:

```
Start
Set totalClasses ← 3

For i ← 1 to totalClasses do
    Print "Attending class", i
End For

Print "All classes attended for the day!"
Stop
```

1.3 Code:

1.3.1 Solving coding problems:

```
import java.util.Scanner;

public class D3_1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter how many coding problems you want to solve: ");
        int totalProblems = sc.nextInt();

        for (int i = 1; i <= totalProblems; i++) {
            System.out.println("Solving problem " + i + "...");
        }
        System.out.println("You have completed " + totalProblems + " problems. Good job!");
    }
}
```

1.3.2 Checking emails:

```
public class D3_1 {
    public static void main(String[] args) {
        int unreadEmails = 5;

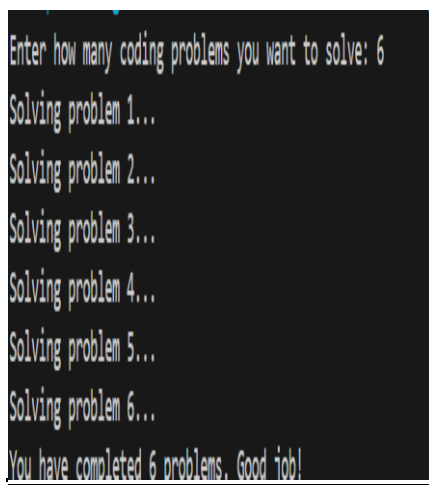
        while (unreadEmails > 0) {
            System.out.println("Reading an email...");
            unreadEmails--;
        }
        System.out.println("Inbox is clear!");
    }
}
```

1.3.3 Attending classes:

```
public class D3_1 {  
    public static void main(String[] args) {  
        int totalClasses = 3;  
  
        for (int i = 1; i <= totalClasses; i++) {  
            System.out.println("Attending class " + i);  
        }  
        System.out.println("All classes attended for the day!");  
    }  
}
```

1.4 Output:

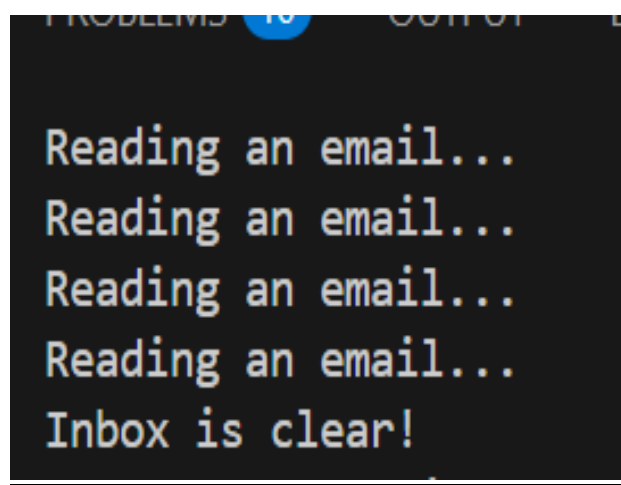
Case 1: Solving coding



A terminal window showing the execution of a program. The user enters '6' for the number of problems. The program then iterates from 1 to 6, printing 'Solving problem 1...' through 'Solving problem 6...'. Finally, it prints 'You have completed 6 problems. Good job!'.

```
Enter how many coding problems you want to solve: 6  
Solving problem 1...  
Solving problem 2...  
Solving problem 3...  
Solving problem 4...  
Solving problem 5...  
Solving problem 6...  
You have completed 6 problems. Good job!
```

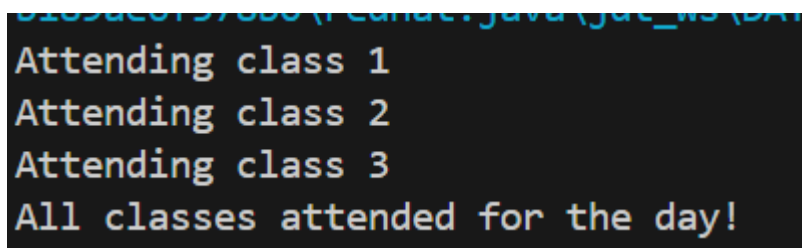
Case 2: Checking emails



A terminal window showing the execution of a program. It prints 'Reading an email...' four times in a loop. After the loop, it prints 'Inbox is clear!'.

```
PROBLEMS TO SOLVE: 6  
Solving problem 1...  
Solving problem 2...  
Solving problem 3...  
Solving problem 4...  
Solving problem 5...  
Solving problem 6...  
You have completed 6 problems. Good job!  
  
Reading an email...  
Reading an email...  
Reading an email...  
Reading an email...  
Inbox is clear!
```

Case 3: Attending classes



A terminal window showing the execution of a program. It prints 'Attending class 1', 'Attending class 2', and 'Attending class 3' in a loop. Finally, it prints 'All classes attended for the day!'.

```
Attending class 1  
Attending class 2  
Attending class 3  
All classes attended for the day!
```

Activity 2: Code Duplication

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

2.1 Algorithm:

Step 1: Print “Hello!”

Step 2: Repeat the print statement 10 times manually

Step 3: Done

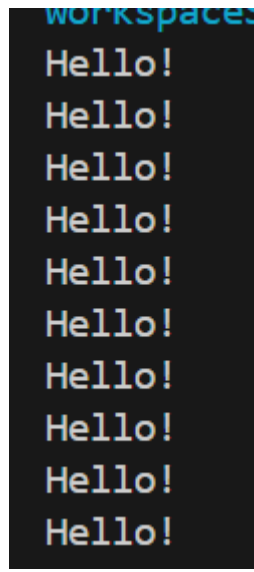
2.2 Pseudocode:

```
PRINT "Hello!"  
PRINT "Hello!"  
PRINT "Hello!"  
PRINT "Hello!"  
PRINT "Hello!"  
PRINT "Hello!"  
PRINT "Hello!"  
PRINT "Hello!"  
PRINT "Hello!"  
PRINT "Hello!"
```

2.3 Code:

```
public class D3_2 {  
    public static void main(String[] args) {  
        System.out.println("Hello!");  
        System.out.println("Hello!");  
        System.out.println("Hello!");  
        System.out.println("Hello!");  
        System.out.println("Hello!");  
        System.out.println("Hello!");  
        System.out.println("Hello!");  
        System.out.println("Hello!");  
        System.out.println("Hello!");  
    }  
}
```

2.4 Output:



```
workspaces
Hello!
Hello!
Hello!
Hello!
Hello!
Hello!
Hello!
Hello!
Hello!
Hello!
```

Activity 3: Countdown Print numbers from 10 to 1, then “Blastoff!”

3.1 Algorithm:

Step 1: Set number = 10

Step 2: While number \geq 1

- a. Print the number
- b. Decrement the number by 1

Step 3: Print "Blastoff!"

3.2 Pseudocode:

```
set number = 10
while number >= 1
    print number
    number = number - 1
end while
print "blastoff!"
```

3.3 Code:

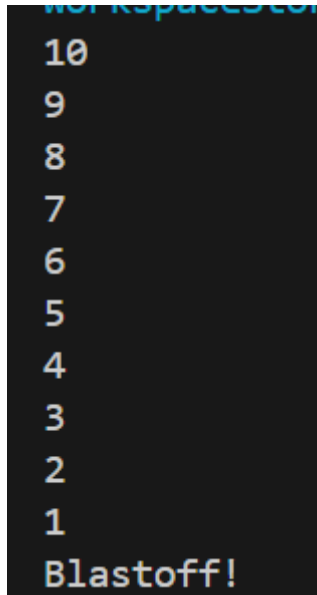
```
public class D3_3{
    public static void main(String[] args) {
        for (int i = 10; i >= 1; i--) {
```

```

        System.out.println(i);
    }
    System.out.println("Blastoff!");
}
}

```

3.4 Output



```

10
9
8
7
6
5
4
3
2
1
Blastoff!

```

Activity 4: Sum Until Zero Ask user for numbers repeatedly until they enter 0. Sum and print the total.

4.1 Algorithm:

Step 1: Initialize sum = 0

Step 2: Repeat:

- a. Ask user to enter a number
- b. If number is 0 → stop
- c. Else → add number to sum

Step 3: Print the total sum

4.2 Pseudocode:

SET sum = 0

REPEAT

 INPUT number

 IF number != 0 THEN

 sum = sum + number

 END IF

```
UNTIL number == 0
PRINT "Total sum is", sum
```

4.3 Code:

```
import java.util.Scanner;

public class D3_4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int number, sum = 0;

        do {
            System.out.print("Enter a number (0 to stop): ");
            number = scanner.nextInt();
            sum += number;
        } while (number != 0);

        System.out.println("Total sum is: " + sum);
    }
}
```

4.4 Output:

**Case1: input of 1
number**

**Case2: input of 5
values until I press
0.**

**Case3: input of 7
values until I press
0.**

```
Enter a number (0 to stop): 4
Enter a number (0 to stop): 0
Total sum is: 4
PS C:\Users\Admin\OneDrive\Desktop
PS C:\Users\Admin\OneDrive\Desktop
PS C:\Users\Admin\OneDrive\Desktop
ridgecourseStemup\Stemup_Bridge_co
X:+ShowCodeDetailsInExceptionMessa
b189ae6f578b0\redhat.java\jdt_ws\
Enter a number (0 to stop): 4
Enter a number (0 to stop): 5
Enter a number (0 to stop): 6
Enter a number (0 to stop): 3
Enter a number (0 to stop): 4
Enter a number (0 to stop): 0
Total sum is: 22
PS C:\Users\Admin\OneDrive\Desktop
PS C:\Users\Admin\OneDrive\Desktop
PS C:\Users\Admin\OneDrive\Desktop
ridgecourseStemup\Stemup_Bridge_co
X:+ShowCodeDetailsInExceptionMessa
b189ae6f578b0\redhat.java\jdt_ws\
Enter a number (0 to stop): 3
Enter a number (0 to stop): 5
Enter a number (0 to stop): 6
Enter a number (0 to stop): 7
Enter a number (0 to stop): 8
Enter a number (0 to stop): 7
Enter a number (0 to stop): 0
Total sum is: 36
PS C:\Users\Admin\OneDrive\Desktop
```

Activity 5: Guess the number. Generate a random number between 1 and 10. Ask user to guess. Provide feedback and loop until correct.

5.1 Algorithm:

Step 1: Generate a random number between 1 and 10 and store in target

Step 2: Repeat:

- a. Ask user to guess a number
- b. If guess < target → print "Too low"
- c. If guess > target → print "Too high"
- d. If guess == target → print "Correct!" and stop

Step 3: End

5.2 Pseudocode:

set target = random number between 1 and 10

repeat

input guess

if guess < target then

print "too low"

else if guess > target then

print "too high"

else

print "correct!"

end if

until guess == target

5.3 Code:


```

import java.util.Scanner;
import java.util.Random;

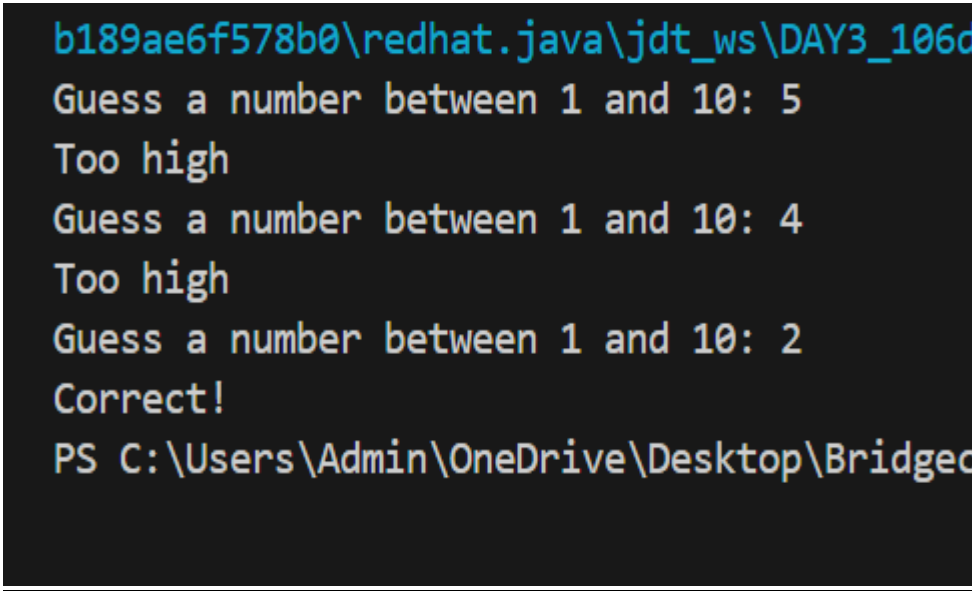
public class D3_5 {
    public static void main(String[] args) {
        Random rand = new Random();
        int target = rand.nextInt(10) + 1;
        Scanner scanner = new Scanner(System.in);
        int guess;

        do {
            System.out.print("Guess a number between 1 and 10: ");
            guess = scanner.nextInt();

            if (guess < target) {
                System.out.println("Too low");
            } else if (guess > target) {
                System.out.println("Too high");
            } else {
                System.out.println("Correct!");
            }
        } while (guess != target);
    }
}

```

5.4 Output:



```

b189ae6f578b0\redhat.java\jdt_ws\DAY3_106d
Guess a number between 1 and 10: 5
Too high
Guess a number between 1 and 10: 4
Too high
Guess a number between 1 and 10: 2
Correct!
PS C:\Users\Admin\OneDrive\Desktop\Bridged

```

Activity 6: Infinite Loop Debugging Analyze and fix:

```

int counter = 0;
while (counter < 5) {
    System.out.println("Hello");
}

```

6.1 Algorithm:

Step1: Initialize counter = 0

Step2: While counter < 5:

- a. Print "Hello"
- b. Increment counter

Step3: End

6.2 Pseudocode:

SET counter = 0

WHILE counter < 5

 PRINT "Hello"

 counter = counter + 1

END WHILE

6.3 Code:

6.3.1 Issue in Code:

```
while (counter < 5) {  
    System.out.println("Hello");  
}
```

- will run forever because counter is never incremented, so counter < 5 is always true.

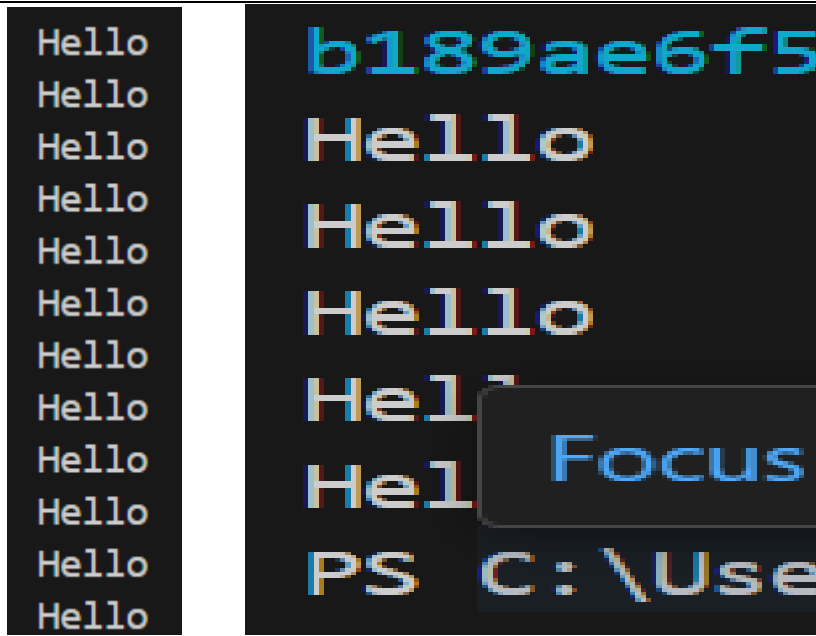
6.3.2 Fixed Code:

```
int counter = 0;  
while (counter < 5) {  
    System.out.println("Hello");  
    counter++; // Fix: increment counter to avoid infinite loop  
}
```

6.4 Output:

Case1: issue in code as its running in loop

Case2: Fixed code after incrementing counter



Activity 7: Even Numbers Print even numbers from 2 to 20 using a for loop.

7.1 Algorithm:

Step1: Start
Step2: Declare an integer variable n and assign the value 20 to it.
Step3: Initialize a loop variable i with value 2.
Step4: Repeat the following steps while i < n:
Step5: Check if i % 2 == 0 (i.e., if i is divisible by 2):
Step6: If yes, then print the value of i.
Step7: Increment the value of i by 1.
Step8: End loop
Step9: End

7.2 Pseudocode:

```
start

set n to 20

for i from 2 to n-1 do
    if i mod 2 equals 0 then
        print i
    endif
endfor

End
```

7.3 Code:

```
public class D3_7 {

    public static void main(String[] args) {

        int n=20;
        for (int i=2; i < n; i++){
            if(i%2==0)
```

```

        System.out.println(i);
    }
}

```

7.4 Output:

```

a59922
2
4
6
8
10
12
14
16
18
20
PS C:\

```

Activity 8: Factorial Calculator Calculate $n!$ for user input n . Handle edge case when $n == 0$.

8.1 Algorithm:

- Step1: Create a scanner object to take user input.
- Step2: Prompt the user: “Enter the number”.
- Step3: Read the integer input and store it in variable fact.
- Step4: Initialize a variable result and set it to 1.
- Step5: For each integer i from 2 to fact, do:
- Step6: Multiply result by i and store the result back in result.
- Step7: End For
- Step8: Print the final value of result (which is the factorial).

8.2 Pseudocode:

```

start
prompt user to enter a number
read fact
set result to 1

for i from 2 to fact do
    result = result * i
endfor

print result
End

```

8.3 Code:

```
import java.util.Scanner;

public class D3_8 {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the number");
        int fact=sc.nextInt();

        int result=1;

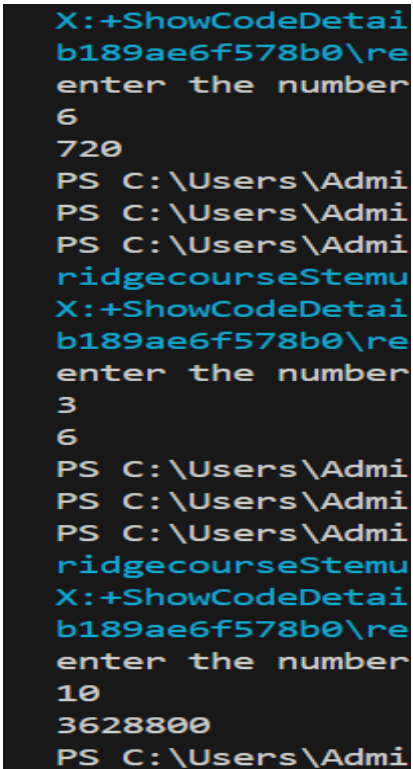
        for(int i=2; i<=fact; i++){
            result = result *i;
        }
        System.out.println(result);
    }
}
```

8.4 Output:

Case1: input as 6

Case2: input as 3

Case4: input as 10



```
X:+ShowCodeDetail
b189ae6f578b0\re
enter the number
6
720
PS C:\Users\Admi
PS C:\Users\Admi
PS C:\Users\Admi
ridgecourseStemu
X:+ShowCodeDetail
b189ae6f578b0\re
enter the number
3
6
PS C:\Users\Admi
PS C:\Users\Admi
PS C:\Users\Admi
ridgecourseStemu
X:+ShowCodeDetail
b189ae6f578b0\re
enter the number
10
3628800
PS C:\Users\Admi
```

Activity 9: Count 'a' in String Ask for a string input. Count how many times 'a' or 'A' appears.

9.1 Algorithm:

Step1: Define a string variable str and assign it the value "javajava".

Step2: Find the total number of characters in str and store it in totalcount.

Step3: Replace all occurrences of 'a' in str with an empty string "", and store the length of the new string in removea.

Step4: Subtract removea from totalcount and store the result in variable count.

Step5: Print count.

9.2 Pseudocode:

start

set str to "javajava"

set totalcount to length of str

set removea to length of str after replacing all 'a' with ""

set count to totalcount - removea

print count

End

9.3 Code:

```
public class Prob23 {  
    public static void main(String[] args) {  
        String str = "javajava";  
        int totalcount= str .length();  
        int removea = str .replace("a","").length();  
        int count=totalcount-removea;  
        System.out.println(count);  
    }  
}
```

9.4: Output

```
X:\ShowCodeDetailsInExceptionMessages -cp C:\User  
b189ae6f578b0\redhat.java\jdt_ws\DAY3_106df28\bin' 'P  
4  
PS C:\Users\Admin\OneDrive\Desktop\BridgecourseStemup
```

Activity 10: Factorial Calculator Calculate $n!$ for user input n .
Handle edge case when $n == 0$.

Algorithm

- Algorithm to Calculate Factorial of a Number
- Start
- Create a scanner object to take user input.
- Prompt the user: "Enter the number".
- Read the integer input and store it in variable fact.
- Initialize a variable result and set it to 1.
- For each integer i from 2 to fact, do:
- Multiply result by i and store the result back in result.
- End For
- Print the final value of result (which is the factorial).
- End

Pseudocode:

1. START
2. Prompt user to enter a number
3. READ fact
4. SET result TO 1
5. FOR i FROM 2 TO fact DO
6. result = result * i
7. ENDFOR
8. PRINT result
9. END

Code:

```
import java.util.Scanner;

public class Prob22 {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int fact=sc.nextInt();

        int result=1;

        for(int i=1; i<=fact; i++){
            result = result * i;
        }
        System.out.println(result);
    }
}
```

Output:

Case1: input as 6

```
ew -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/sariyamazha
774c2/redhat.java/jdt_ws/StemUp\ BridgeCourse_994dd6f4/bin D3_8
enter the number
6
720
```

Case2: input as 3

```
(base) sariyamazhar@SARIYAs-Air StemUp BridgeCourse % cd /Users/
hines/jdk-24.jdk/Contents/Home/bin/java --enable-preview -XX:+Sho
Code/User/workspaceStorage/470f142dc3020e2428b4528a3fc774c2/redha
enter the number
3
6
```

Case3: input as
10

```
(base) sariyamazhar@SARIYAs-Air StemUp BridgeCourse % cd /Users/
hines/jdk-24.jdk/Contents/Home/bin/java --enable-preview -XX:+Sho
Code/User/workspaceStorage/470f142dc3020e2428b4528a3fc774c2/redha
enter the number
10
3628800
(base) sariyamazhar@SARIYAs-Air StemUp BridgeCourse % █
```

Activity 11: Simple Star Pattern Print: * Using one for loop.**

11.1 Algorithm:

Step1: Initialize a loop variable i with value 1.

Step2: Repeat the following steps while $i \leq 5$:
Step3: Print "*" without moving to the next line.
Step4: Increment i by 1.
Step5: End loop

11.2 Pseudocode:

START

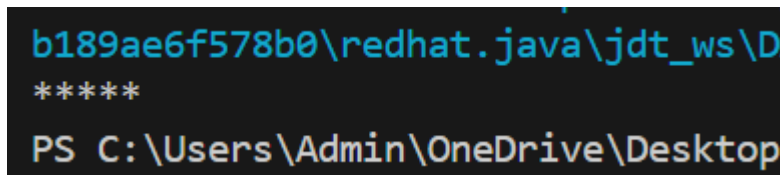
FOR i FROM 1 TO 5 DO
 PRINT "*", without newline
ENDFOR

END

11.3 Code:

```
public class D3_10 {  
    public static void main(String[] args){  
        for (int i=1; i <=5; i++){  
            System.out.print("*");  
        }  
    }  
}
```

11.4 Output:



```
b189ae6f578b0\redhat.java\jdt_ws\D  
*****  
PS C:\Users\Admin\OneDrive\Desktop
```

Activity 12: Prime Checker Check if a number is prime using a loop and break.

12.1 Algorithm:

Problem 3.11: Prime Checker Check if a number is prime using a loop and break.

Step 1: Start
Step 2: Input a number no from the user
Step 3: Initialize a boolean variable flag to true
Step 4: If no is less than or equal to 1, set flag to false
Step 5: Loop from $i = 2$ to $i < no$
Step 6: If $no \% i == 0$, set flag = false and break the loop
Step 7: If flag == true, print "Prime number"
Step 8: Else, print "Not a prime"
Step 9: Stop

12.2 Pseudocode:

Start
Read no
Set flag \leftarrow true

```

If  $no \leq 1$  then
    flag  $\leftarrow$  false
Else
    For i  $\leftarrow$  2 to no - 1 do
        If no mod i = 0 then
            flag  $\leftarrow$  false
            Break
        End If
    End For
End If

If flag = true then
    Print "Prime number: ", no
Else
    Print "Not a prime: ", no
End If
Stop

```

12.3 Code:

```

import java.util.Scanner;
public class D3_1 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number");
        int no=sc.nextInt();
        boolean flag=true;

        for(int i=2;i<no;i++){
            if(no%i==0){
                flag=false;
                break;
            }
        }
        if(flag==true){
            System.out.println("Prime number: "+no);
        }else{
            System.out.println("not a prime: "+no);
        }
    }
}

```

12.4 Output:

Case1: input as
2

Case2: input
as 88

Case3: input
as 222

```
6189ae6f578b0\redhat.java\jdt_ws\DAY3_106d
Enter a number
2
Prime number: 2
PS C:\Users\Admin\OneDrive\Desktop\Bridgec
PS C:\Users\Admin\OneDrive\Desktop\Bridgec
PS C:\Users\Admin\OneDrive\Desktop\Bridgec
ridgecourseStemup\Stemup_Bridge_course\DAY
X:+ShowCodeDetailsInExceptionMessages' '-c
b189ae6f578b0\redhat.java\jdt_ws\DAY3_106d
Enter a number
88
not a prime: 88
PS C:\Users\Admin\OneDrive\Desktop\Bridgec
PS C:\Users\Admin\OneDrive\Desktop\Bridgec
PS C:\Users\Admin\OneDrive\Desktop\Bridgec
ridgecourseStemup\Stemup_Bridge_course\DAY
X:+ShowCodeDetailsInExceptionMessages' '-c
b189ae6f578b0\redhat.java\jdt_ws\DAY3_106d
Enter a number
222
not a prime: 222
PS C:\Users\Admin\OneDrive\Desktop\Bridgec
```

Activity 13: Skip Negatives Input 5 numbers. Use continue to skip negative ones and sum the rest.

13.1 Algorithm:

Step 1: Start
Step 2: Initialize sum to 0
Step 3: Repeat the following steps 5 times
Step 4: Read a number num
Step 5: If $\text{num} < 0$, skip this iteration using continue
Step 6: Otherwise, add num to sum
Step 7: After the loop ends, print the value of sum
Step 8: Stop

13.2 Pseudocode:

```
Start
Set sum  $\leftarrow$  0

Repeat 5 times:
    Read num
    If  $\text{num} < 0$  then
        Continue to next iteration
    End If
    sum  $\leftarrow$  sum + num
End Repeat

Print "Sum of non-negative numbers: ", sum
Stop
```

13.3 Code:

```
import java.util.Scanner;

public class D3_12 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int sum = 0;
        System.out.println("Enter 5 numbers:");
        for (int i = 1; i <= 5; i++) {
            int num = sc.nextInt();

            if (num < 0) {
                continue;
            }
            sum += num;
        }

        System.out.println("Sum of non-negative numbers: " + sum);
    }
}
```

13.4 Output:

**Case1: All
positive integers**

**Case2: Different
positive integers**

**Case3: positive
with negative**

```
7
8
9
6
5
Sum of non-negative numbers: 35
PS C:\Users\Admin\OneDrive\Desktop>
PS C:\Users\Admin\OneDrive\Desktop>
PS C:\Users\Admin\OneDrive\Desktop>
ridgecourseStemup\Stemup_Bridge_
X:+ShowCodeDetailsInExceptionMes
b189ae6f578b0\redhat.java\jdt_ws
Enter 5 numbers:
100
777
777
54
6
78
Sum of non-negative numbers: 101
PS C:\Users\Admin\OneDrive\Desktop>
PS C:\Users\Admin\OneDrive\Desktop>
PS C:\Users\Admin\OneDrive\Desktop>
ridgecourseStemup\Stemup_Bridge_
X:+ShowCodeDetailsInExceptionMes
b189ae6f578b0\redhat.java\jdt_ws
Enter 5 numbers:
-1
-2
4
-7
7
Sum of non-negative numbers: 11
Indexing completed. Java: Ready
```

Activity 14: Pyramid Pattern Challenge Input height. Print centered pyramid.

14.1 Algorithm:

- Step 1: Start
- Step 2: Input the height of the pyramid
- Step 3: Repeat the following steps for i from 1 to height
- Step 4: Print height - i number of spaces
- Step 5: Print 2 * i - 1 number of * characters
- Step 6: Move to the next line
- Step 7: End outer loop
- Step 8: Stop

14.2 Pseudocode:

```
Start
Read height
For i ← 1 to height do
    For j ← 1 to height - i do
        Print " "
    End For
```

```

    For k ← 1 to (2 * i - 1) do
        Print "*"
    End For
    Print new line
End For
Stop

```

14.3 Code:

```

import java.util.Scanner;

public class D3_15 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter height of the pyramid: ");
        int height = sc.nextInt();

        for (int i = 1; i <= height; i++) {

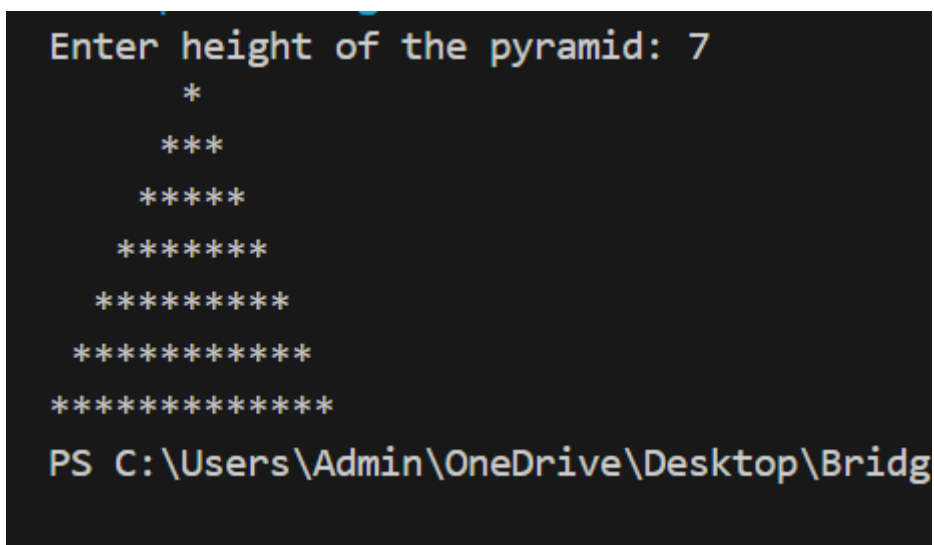
            for (int j = 1; j <= height - i; j++) {
                System.out.print(" ");
            }

            for (int k = 1; k <= (2 * i - 1); k++) {
                System.out.print("*");
            }
            System.out.println();
        }

        sc.close();
    }
}

```

14.4 Output:



```

Enter height of the pyramid: 7
      *
     ***
    *****
   ********
  *********
 ***
*****
*****
PS C:\Users\Admin\OneDrive\Desktop\Bridg

```

Activity 14: Rectangle Pattern Input rows and cols, print a rectangle of *.

14.1 Algorithm:

Step 1: Start
Step 2: Input the number of rows and cols
Step 3: Repeat the following steps for i from 1 to rows
Step 4: Repeat the following steps for j from 1 to cols
Step 5: Print "*" without newline
Step 6: After inner loop ends, print newline to move to next row
Step 7: End outer loop
Step 8: Stop

14.2 Pseudocode:

```
Start
Read rows
Read cols

For i ← 1 to rows do
    For j ← 1 to cols do
        Print "*"
    End For
    Print new line
End For

Stop
```

14.2 Code:

```
import java.util.Scanner;

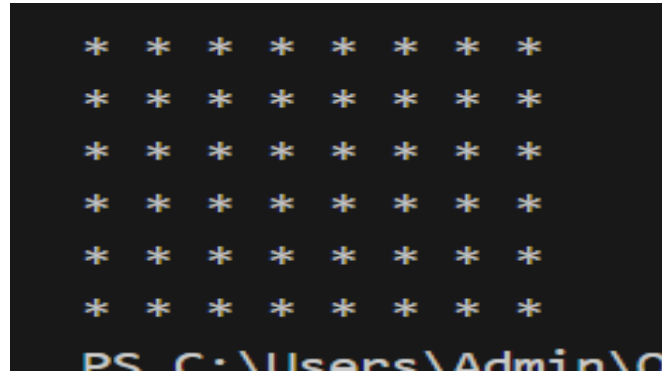
public class D3_13 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of rows: ");
        int rows = sc.nextInt();

        System.out.print("Enter number of columns: ");
        int cols = sc.nextInt();

        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= cols; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

14.3 Output:



Activity 15: Triangle Pattern where the user inputs the height and the program prints a right-angled triangle made of *.

15.1 Algorithm:

- Step 1: Start
- Step 2: Input the height of the triangle
- Step 3: Repeat the following steps for i from 1 to height
- Step 4: Repeat the following steps for j from 1 to i
- Step 5: Print "*" without newline
- Step 6: After inner loop ends, print newline to move to next row
- Step 7: End outer loop
- Step 8: Stop

15.2 Pseudocode:

```
Start
Read height
For i ← 1 to height do
    For j ← 1 to i do
        Print "*"
    End For
    Print new line
End For
Stop
```

15.3 Code:

```
import java.util.Scanner;

public class D3_14 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

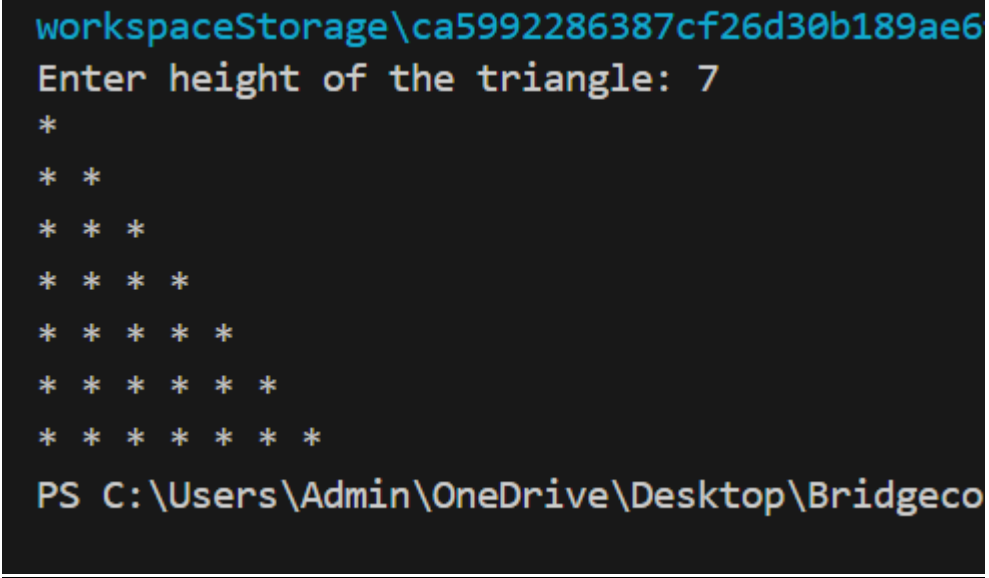
        System.out.print("Enter height of the triangle: ");
        int height = sc.nextInt();

        for (int i = 1; i <= height; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
        }
    }
}
```



```
    }  
    System.out.println();  
}  
  
sc.close();  
}  
}
```

15.4 Output:



```
workspaceStorage\ca5992286387cf26d30b189ae6  
Enter height of the triangle: 7  
*  
* *  
* * *  
* * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
PS C:\Users\Admin\OneDrive\Desktop\Bridgeco
```