

SECTION 1: Error-Driven Learning Assignment: Loop Errors

1.Snippet:

// Error to investigate: Why does this loop run infinitely? How should the loop control variable be adjusted?

The infinite loop occurs because the loop control variable `i` is decremented (`i--`) instead of incremented (`i++`) in the for loop. The condition `i < 10` is always true, causing the loop to run indefinitely.

To correct the code, simply change the decrement operator (`i--`) to an increment operator (`i++`) in the for loop.

```
public class FiniteForLoop {  
    public static void main(String[] args) {  
        for (int i = 0; i < 10; i++) { // Change i-- to i++  
            System.out.println(i);  
        }  
    }  
}
```

With this correction, the loop will iterate from `i = 0` to `i = 9`, printing the numbers 0 through 9, and then terminate as intended.

2.Snippet:

// Error to investigate: Why does the loop not execute as expected? What is the issue with the condition in the while loop?

The issue lies in the condition of the while loop: `count = 0`. This is an assignment operator (`=`), not a comparison operator (`==`). As a result, the condition is always true because the assignment `count = 0` is successful. The initial value of `count` is 5, and the loop decrements `count` (`count--`) until it reaches 0.

We have to change the assignment operator (`=`) to a comparison operator (`==`) in the while loop condition.

And then update the initial value of `count` or the loop condition to ensure the loop executes as intended.

```

public class CorrectedWhileCondition {
    public static void main(String[] args) {
        int count = 5;
        while (count >= 0) { // Update the condition
            System.out.println(count);
            count--;
        }
    }
}

```

corrected output is 5 to 1.

3.Snippet:

// Error to investigate: Why does the loop only execute once? What is wrong with the loop condition in the do while loop?

The problem is that the loop condition is not correct for the intended behavior. The loop will continue to execute as long as num is greater than 0. Since num is incremented inside the loop, it will eventually exceed any desired limit.

To correct the code, update the loop condition to reflect the desired termination criterion. E.g if you want the loop to execute 5 times, you can change the condition to `num < 5`:

```

public class DoWhileCorrectedCondition {
    public static void main(String[] args) {
        int num = 0;
        do {
            System.out.println(num);
            num++;
        } while (num < 5); // Update the condition
    }
}

```

corrected output is 0 to 4.

4.Snippet:

// Error to investigate: What is the issue with the loop boundaries? How should the loop be adjusted to meet the expected output?

The issue lies in the loop boundaries of the for loop. The loop starts from $i = 1$ and continues until $i \leq 10$. This means the loop will iterate 10 times, printing numbers 1 to 10.

To correct the code, you should adjust the loop boundaries to meet the expected output. Since the task expected only numbers 1 to 9, you can change the loop condition to $i < 10$:

```
public class CorrectedForLoop {  
    public static void main(String[] args) {  
        for (int i = 1; i < 10; i++) { // Adjust the loop condition  
            System.out.println(i);  
        }  
    }  
}
```

This corrected version will output the numbers 1 to 9.

5.Snippet:

// Error to investigate: Why does this loop not print numbers in the expected order? What is the problem with the initialization and update statements in the `for` loop?

The issue lies in the initialization and update statements of the for loop. The loop starts from $i = 10$ and decrements i by 1 in each iteration ($i--$). However, the loop condition is $i \geq 0$, which means the loop will continue until i becomes -1.

The task expected numbers in the order from 10 to 0, you can keep the initialization as $i = 10$ but update the loop condition to $i \geq 0$ and keep the decrement statement as $i--$.

However, to print numbers from 10 to 0, including 0, you should change the loop condition to $i > -1$ or simply $i \geq 0$.

```
public class CorrectedForLoop {  
    public static void main(String[] args) {
```

```

        for (int i = 10; i >= 0; i--) {
            System.out.println(i);
        }
    }
}

```

Corrected output is 10 to 0.

6.Snippet:

// Error to investigate: Why does "Done" print only once, outside the loop? How should the loop body be enclosed to include all statements within the loop.

The issue lies in the placement of the loop body. In this a single statement follows a for loop declaration, it is considered the loop body. In this case, the loop body is only the first System.out.println(i) statement.

To correct the code, you should enclose the loop body within curly brackets {} to include all statements within the loop:

```

public class CorrectedForLoop {
    public static void main(String[] args) {
        for (int i = 0; i < 5; i++) {
            System.out.println(i);
            System.out.println("Done");
        }
    }
}

```

Corrected output is 0 to 4, followed by "Done" on each line, is the expected output.

7.Snippet:

// Error to investigate: Why does this code produce a compilation error? What needs to be done to initialize the loop variable properly?

The issue lies in the uninitialized variable count. In Java, local variables (like count) must be initialized before they can be used. Since count is not initialized, the compiler throws an error

when it encounters the while loop condition `count < 10`

To correct the code, you need to initialize the count variable before the while loop. You can do this by assigning a value to count, like so:

```
public class InitializedWhileLoop {  
    public static void main(String[] args) {  
        int count = 0; // Initialize count to 0  
        while (count < 10) {  
            System.out.println(count);  
            count++;  
        }  
    }  
}
```

By initializing count to 0, the code can now compile and run without errors. The while loop will iterate from 0 to 9.

8.Snippet:

// Error to investigate: Why does this loop print unexpected numbers? What adjustments are needed to print the numbers from 1 to 5?

The issue lies in the loop condition and the update statement. The loop starts with `num = 1` and decrements `num` by 1 in each iteration (`num--`). However, the loop condition is `num > 0`, which means the loop will continue until `num` becomes 0. As a result, the loop will print only the number

To correct the code and print numbers from 1 to 5:

1. Initialize `num` to 5 instead of 1.
2. Update the loop condition to `num >= 1` to ensure the loop runs from 5 to 1.
3. Keep the update statement as `num--` to decrement `num` by 1 in each iteration.

```
public class CorrectedDoWhileLoop {  
    public static void main(String[] args) {
```

```

int num = 5; // Initialize num to 5
do {
    System.out.println(num);
    num--; // Decrement num by 1
} while (num >= 1); // Update the loop condition
}
}

```

Corrected is output 5 to 1.

9.Snippet:

// Error to investigate: Why does the loop print unexpected results or run infinitely? How should the loop update expression be corrected?

The issue lies in the loop update expression `i += 2`. This expression increments `i` by 2 in each iteration. However, the loop condition is `i < 5`, which means the loop will terminate when `i` reaches 5.

Unexpected Behavior of the loop will print the numbers 0, 2, and 4. Although the loop doesn't run infinitely, it doesn't print all numbers from 0 to 4 as one might expect.

To correct the loop and print all numbers from 0 to 4, you should update the loop update expression to `i++`, which increments `i` by 1 in each iteration:

```

public class CorrectedForLoop {
    public static void main(String[] args) {
        for (int i = 0; i < 5; i++) { // Update the loop update expression
            System.out.println(i);
        }
    }
}

```

Corrected output is 0 to 4.

10.Snippet:

// Error to investigate: Why does the loop execute indefinitely? What is wrong with the loop condition?

The issue lies in the loop condition `num = 10`. This is an assignment operator (`=`), not a comparison operator (`==`). As a result, the condition is always true because the assignment `num = 10` is successful, and the loop executes indefinitely.

To correct the code, you should update the loop condition to use a comparison operator (`==`) instead of an assignment operator (`=`). To print numbers from 10 down to a certain value, you should also update the condition to `num >= 0` or any other desired termination value.

```
public class CorrectedWhileLoop {  
    public static void main(String[] args) {  
        int num = 10;  
        while (num >= 0) { // Update the loop condition  
            System.out.println(num);  
            num--;  
        }  
    }  
}
```

Corrected output 10 to 0.

11.Snippet:

// Error to investigate: What will be the output of this loop? How should the loop variable be updated to achieve the desired result?

The issue lies in the loop update statement `i += 2`. This statement increments `i` by 2 in each iteration. However, the loop condition is `i < 5`, which means the loop will terminate when `i` reaches 5.

Unexpected Behavior is the loop will print the numbers 0, 2, and 4. Although the loop doesn't run infinitely, it doesn't print all numbers from 0 to 4 as one might expect.

To correct the loop and print all numbers from 0 to 4, you should update the loop update expression to `i++`, which increments `i` by 1 in each iteration:

```
public class CorrectedLoop {
```

```

public static void main(String[] args) {
    int i = 0;
    while (i < 5) {
        System.out.println(i);
        i++; // Update the loop update expression
    }
}

```

Corrected output is 0 to 4.

12.Snippet:

// Error to investigate: Why does the variable 'x' cause a compilation error? How does scope.

The issue lies in the scope of the variable x. The variable x is declared inside the for loop, which means its scope is limited to the loop body. Once the loop finishes executing, the variable x is no longer accessible.

To correct the code, declare the variable x outside the for loop, so its scope is not limited to the loop body.

```

public class CorrectedLoopVariableScope {
    public static void main(String[] args) {
        int x; // Declare 'x' outside the loop
        for (int i = 0; i < 5; i++) {
            x = i * 2;
        }
        System.out.println(x); // Now 'x' is accessible
    }
}

```

This corrected output is the value of x after the loop finishes executing.

SECTION 2: Guess the Output

1. Snippet: Dry run

Step 1: Initialize Variables

- i is initialized to 1.

- j is initialized to 1.

Step 2: First Iteration of Outer Loop (i = 1)

- Inner loop iterates from j = 1 to j = 2.

- Prints: 1 1 1 2

Step 3: Second Iteration of Outer Loop (i = 2)

- Inner loop iterates from j = 1 to j = 2.

- Prints: 2 1 2 2

Step 4: Third Iteration of Outer Loop (i = 3)

- Inner loop iterates from j = 1 to j = 2.

- Prints: 3 1 3 2

The final answer is:

1 1 1 2

2 1 2 2

3 1 3 2

2.Snippet : Dry run

Step 1: Initialize Variables

- total is initialized to 0.

- i is initialized to 5.

Step 2: First Iteration of Loop (i = 5)

- total becomes $0 + 5 = 5$.

- Since i is not equal to 3, total becomes $5 - 1 = 4$.

Step 3: Second Iteration of Loop ($i = 4$)

- total becomes $4 + 4 = 8$.
- Since i is not equal to 3, total becomes $8 - 1 = 7$.

Step 4: Third Iteration of Loop ($i = 3$)

- total becomes $7 + 3 = 10$.
- Since i is equal to 3, the continue statement is executed, skipping the rest of the current iteration.

Step 5: Fourth Iteration of Loop ($i = 2$)

- total becomes $10 + 2 = 12$.
- Since i is not equal to 3, total becomes $12 - 1 = 11$.

Step 6: Fifth Iteration of Loop ($i = 1$)

- total becomes $11 + 1 = 12$.
- Since i is not equal to 3, total becomes $12 - 1 = 11$.

The final answer is: 11

3.Snippet : Dry run

Step 1: Initialize Variables

- count is initialized to 0.

Step 2: First Iteration of Loop (count = 0)

- count is printed: 0
- count is incremented to 1
- Since count is not equal to 3, the loop continues

Step 3: Second Iteration of Loop (count = 1)

- count is printed: 1
- count is incremented to 2
- Since count is not equal to 3, the loop continues

Step 4: Third Iteration of Loop (count = 2)

- count is printed: 2
- count is incremented to 3
- Since count is equal to 3, the break statement is executed, exiting the loop

Step 5: Print Final Value of count

- count is printed: 3

The final answer is:

0 1 2 3

4.Snippet : Dry run

Step 1: Initialize Variables

- i is initialized to 1.

Step 2: First Iteration of Loop (i = 1)

- i is printed: 1
- i is incremented to 2
- Since i (2) is less than 5, the loop continues

Step 3: Second Iteration of Loop (i = 2)

- i is printed: 2
- i is incremented to 3
- Since i (3) is less than 5, the loop continues

Step 4: Third Iteration of Loop (i = 3)

- i is printed: 3
- i is incremented to 4
- Since i (4) is less than 5, the loop continues

Step 5: Fourth Iteration of Loop (i = 4)

- i is printed: 4
- i is incremented to 5

- Since i (5) is not less than 5, the loop exits

Step 6: Print Final Value of i

- i is printed: 5

The final answer is:

1 2 3 4 5

5.Snippet: Dry run

Step 1: Initialize Variables

- num is initialized to 1.

- i is initialized to 1.

Step 2: First Iteration of Loop ($i = 1$)

- Since i (1) is odd, num becomes $1 - 1 = 0$.

Step 3: Second Iteration of Loop ($i = 2$)

- Since i (2) is even, num becomes $0 + 2 = 2$.

Step 4: Third Iteration of Loop ($i = 3$)

- Since i (3) is odd, num becomes $2 - 3 = -1$.

Step 5: Fourth Iteration of Loop ($i = 4$)

- Since i (4) is even, num becomes $-1 + 4 = 3$.

The final answer is:

3

6.Snippet: Dry run

Step 1: Initialize Variables

- x is initialized to 5.

- y is initialized (but not assigned a value yet).

Step 2: Evaluate Expression

- ++x increments x to 6 and returns 6.
- x-- returns the current value of x (6) and then decrements x to 5.
- --x decrements x to 4 and returns 4.
- x++ returns the current value of x (4) and then increments x to 5.

Step 3: Calculate y

- y is assigned the result of the expression: $6 - 6 + 4 + 4 = 8$.

The final answer is:

8

7.Snippet: Dry run

Step 1: Initialize Variables

- a is initialized to 10.
- b is initialized to 5.

Step 2: Evaluate Expression

- ++a increments a to 11 and returns 11.
- b-- returns the current value of b (5) and then decrements b to 4.
- --a decrements a to 10 and returns 10.
- b++ returns the current value of b (4) and then increments b to 5.

Step 3: Calculate result

- result is assigned the result of the expression: $(11 * 5) - 10 + 4 = 55 - 10 + 4 = 49$.

The final answer is:

49

8.Snippet: Dry run

Step 1: Initialize Variables

- count is initialized to 0.

- i is initialized to 0.

Step 2: First Iteration of Loop (i = 0)

- i++ returns 0 and increments i to 1.

- ++i increments i to 2 and returns 2.

- count becomes $0 + (0 - 2) = -2$.

Step 3: Second Iteration of Loop (i = 2)

- i++ returns 2 and increments i to 3.

- ++i increments i to 4 and returns 4.

- count becomes $-2 + (2 - 4) = -4$.

Step 4: Third Iteration of Loop (i = 4)

- Since i (4) is not less than 4, the loop exits.

The final answer is:

-4

SECTION 3: Lamborghini Exercise:

/*

1. Write a program to calculate the sum of the first 50 natural numbers.

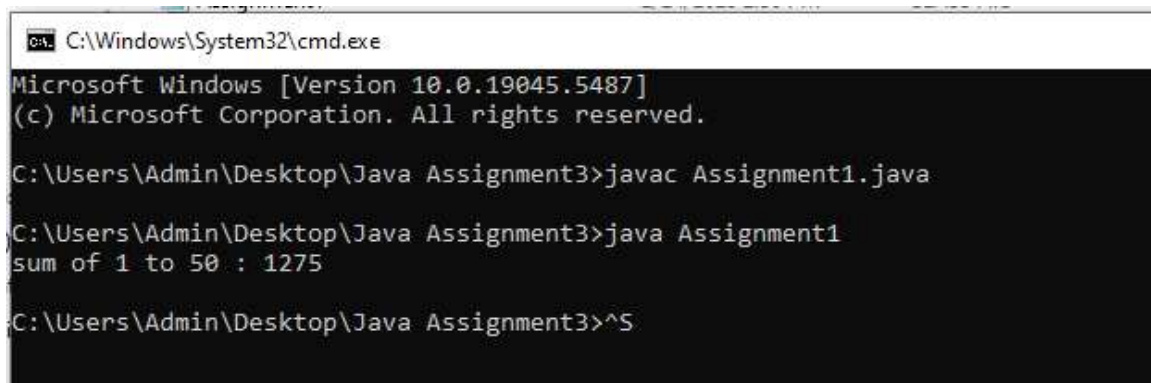
*/

```
class Assignment1{
public static void main(String[] args){
    int sum = 0;
    for(int i=1; i<=50; i++){
        sum+=i;
    }
}
```

```

        System.out.println("sum of 1 to 50 : "+ sum);
    }
}

```



The screenshot shows a Windows Command Prompt window with the title bar 'C:\Windows\System32\cmd.exe'. The text inside the window is as follows:

```

Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment1.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment1
sum of 1 to 50 : 1275

C:\Users\Admin\Desktop\Java Assignment3>^S

```

```

/*
2. Write a program to compute the factorial of the number 10.
*/
class Assignment2{
    public static void main(String[] args){
        int fact = 1;
        for(int i=1; i<=10; i++){
            fact = fact * i;
        }
        System.out.println("Factorial of 10 is "+ fact);
    }
}

```

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment2.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment2
Factorial of 10 is 3628800

C:\Users\Admin\Desktop\Java Assignment3>_

```

/*

3. Write a program to print all multiples of 7 between 1 and 100

*/

```

class Assignment3{
    public static void main(String[] args){
        for (int i = 1; i <= 100; i++) {
            if (i % 7 == 0) {
                System.out.println(i);    }
        }
    }
}

```



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment3.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment3
7
14
21
28
35
42
49
56
63
70
77
84
91
98

C:\Users\Admin\Desktop\Java Assignment3>
```

/*

4. Write a program to reverse the digits of the number 1234. The output should be 4321

*/

```
class Assignment4{
public static void main(String[] args){
    int n = 1234;
    int temp = n;
    int rev = 0;
    while(temp>0){
        int rem = temp%10;
        rev = rev*10 + rem;
        temp = temp/10;
    }
}
```

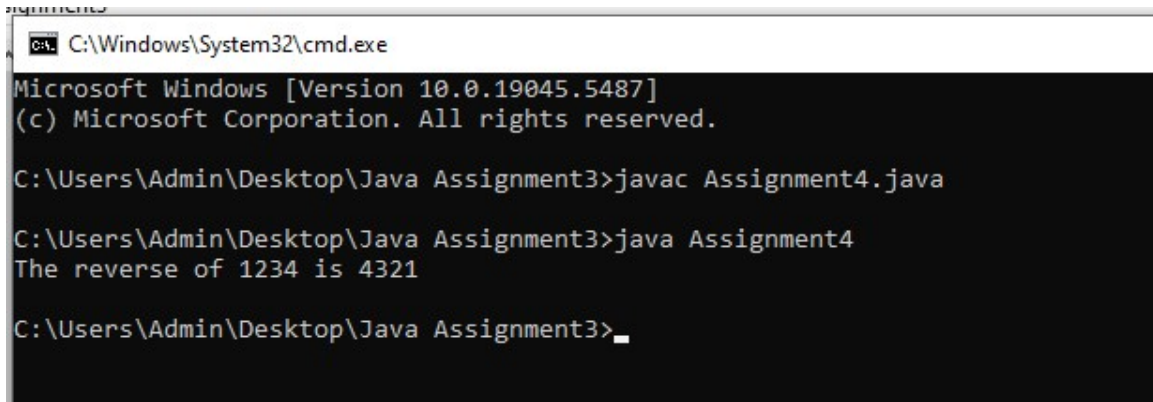
```

    }

    System.out.println("The reverse of "+n+ " is "+rev);

}
}

```



The screenshot shows a Windows command prompt window with the title bar "C:\Windows\System32\cmd.exe". The text inside the window is as follows:

```

Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment4.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment4
The reverse of 1234 is 4321

C:\Users\Admin\Desktop\Java Assignment3>_

```

```

/*

```

5. Write a program to print the Fibonacci sequence up to the number 21.

```

*/

```

```

class Assignment5{

    public static void main(String[] args){

        int n = 21;

        int a = 0;

        int b = 1;

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

            System.out.print(a+ " ");

            int c = a+b;

            a = b;

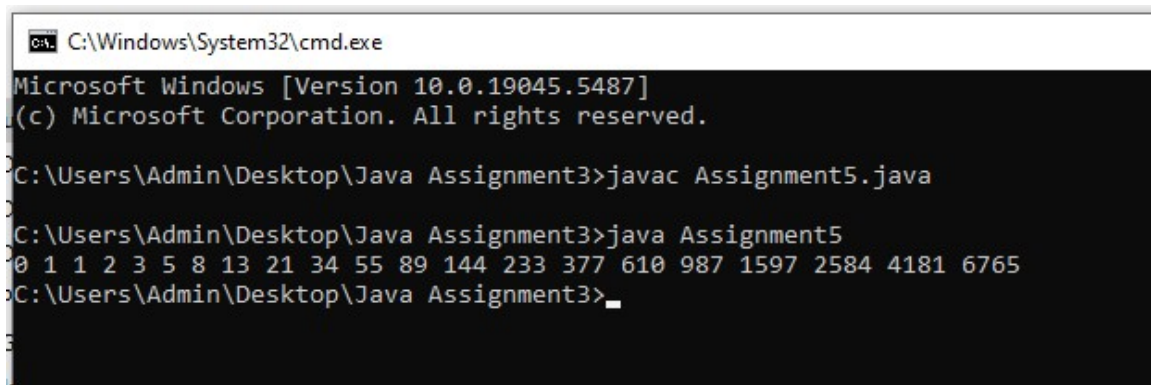
            b = c;

        }

    }
}

```

```
}  
}
```



```
C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 10.0.19045.5487]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\Admin\Desktop\Java Assignment3>javac Assignment5.java  
  
C:\Users\Admin\Desktop\Java Assignment3>java Assignment5  
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765  
C:\Users\Admin\Desktop\Java Assignment3>_
```

```
/*
```

6. Write a program to find and print the first 5 prime numbers.

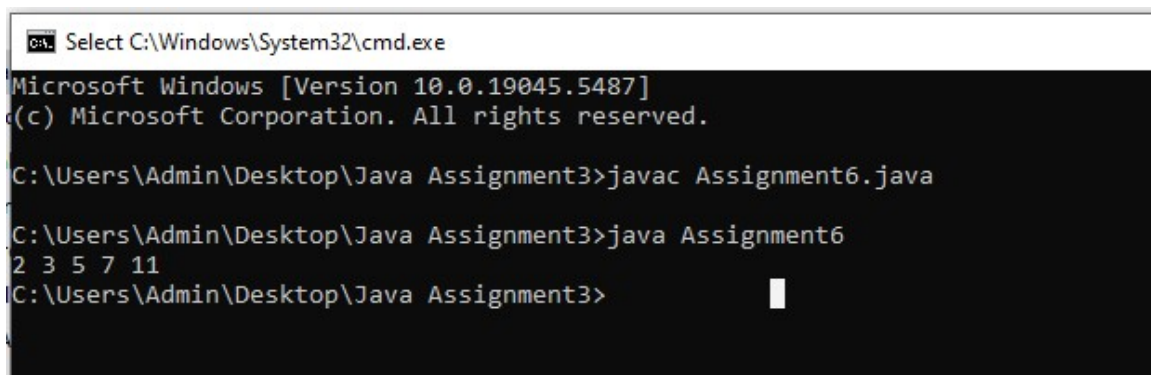
```
*/
```

```
class Assignment6{  
    public static boolean isPrime(int n) {  
        if (n < 2) return false;  
        for (int i = 2; i * i <= n; i++) {  
            if (n % i == 0) return false;  
        }  
        return true;  
    }  
  
    public static void main(String[] args){  
        int count = 0;  
        int num = 2;  
  
        while(count<5){
```

```

        if(isPrime(num)){
            System.out.print(num + " ");
            count++;
        }
        num++;
    }
}
}

```



The screenshot shows a Windows Command Prompt window titled "C:\ Windows\ System32\ cmd.exe". The prompt displays the following text:

```

Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment6.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment6
2 3 5 7 11

C:\Users\Admin\Desktop\Java Assignment3>

```

/*

7. Write a program to calculate the sum of the digits of the number 9876. The output should be

30 (9 + 8 + 7 + 6).

*/

```

class Assignment7{
    public static void main(String[] args){
        int n = 9876;
        int temp = n;
        int sum = 0;
        while(temp>0){
            int rem = temp%10;

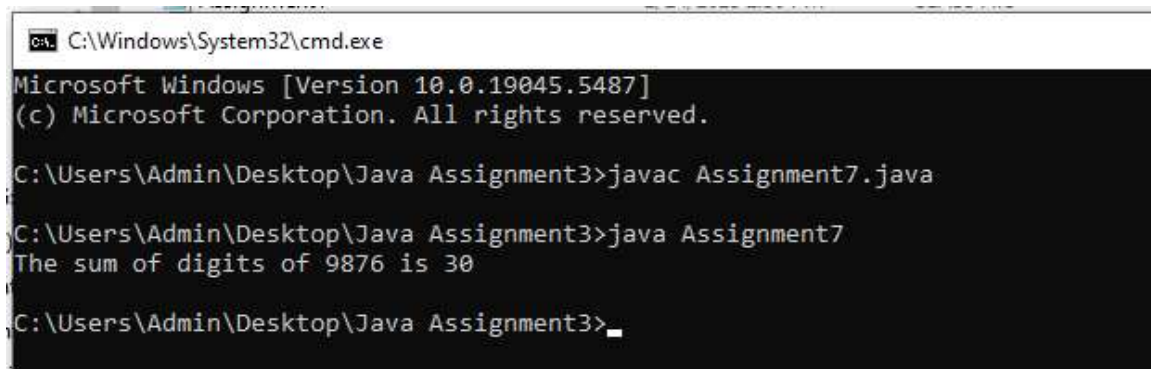
```

```

        sum+=rem;
        temp = temp/10;
    }

    System.out.println("The sum of digits of "+n+ " is "+sum);
}
}

```



The screenshot shows a Windows Command Prompt window with the title bar 'C:\Windows\System32\cmd.exe'. The text inside the window is as follows:

```

Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment7.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment7
The sum of digits of 9876 is 30

C:\Users\Admin\Desktop\Java Assignment3>_

```

/*

8. Write a program to count down from 10 to 0, printing each number.

*/

```

class Assignment8{
    public static void main(String[] args){
        int n = 10;
        for(int i=n; i>=0; i--){
            System.out.println(i);
        }
    }
}

```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment8.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment8
10
9
8
7
6
5
4
3
2
1
0

C:\Users\Admin\Desktop\Java Assignment3>_
```

/*

9. Write a program to find and print the largest digit in the number 4825.

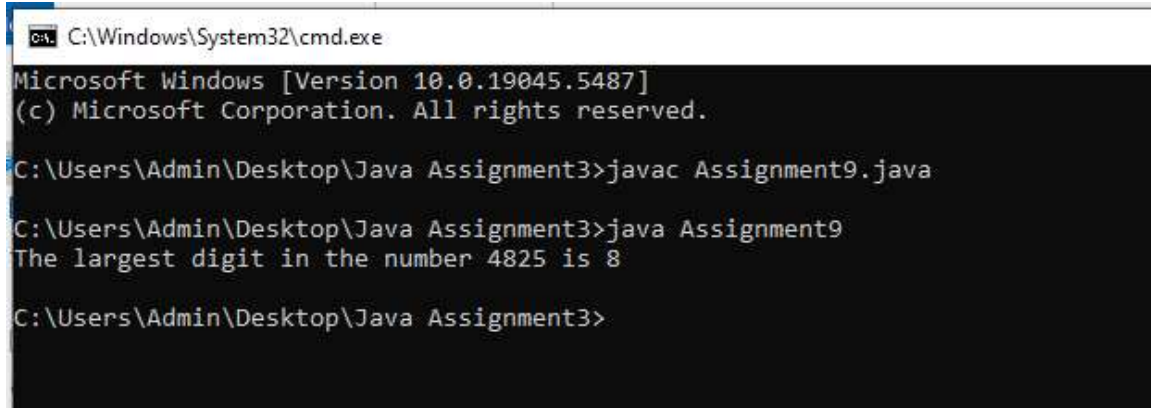
*/

```
class Assignment9{
    public static void main(String[] args){
        int n = 4825;
        int temp = n;
        int large = 0;
        while(temp>0){
            int rem = temp%10;
            if(large<rem){
                large = rem;
            }
            temp = temp/10;
        }
    }
}
```

```

        System.out.println("The largest digit in the number "+n+ " is "+large);
    }
}

```



```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment9.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment9
The largest digit in the number 4825 is 8

C:\Users\Admin\Desktop\Java Assignment3>

```

```

/*
10. Write a program to print all even numbers between 1 and 50.
*/

class Assignment10{
    public static void main(String[] args){
        int n = 50;
        for(int i=1; i<=n; i++){
            if(i%2==0){
                System.out.print(i+" ");
            }
        }
    }
}

```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment10.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment10
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50
C:\Users\Admin\Desktop\Java Assignment3>_
```

/*

11. Write a Java program to demonstrate the use of both pre-increment and post-decrement

operators in a single expression

*/

```
class Assignment11{
    public static void main(String[] args){
        int i = 10;
        int ans = ++i + i--;
        System.out.println("if i=10 then ++i + i-- = "+ans);
    }
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment11.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment11
if i=10 then ++i + i-- = 22

C:\Users\Admin\Desktop\Java Assignment3>_
```



```
/*
```

12. Write a program to draw the following pattern:

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
*/
```

```
class Assignment12{
```

```
public static void main(String[] args){
```

```
    int n = 5;
```

```
    for(int i=1; i<=n; i++){
```

```
        for(int j=1; j<=n; j++){
```

```
            System.out.print("* ");
```

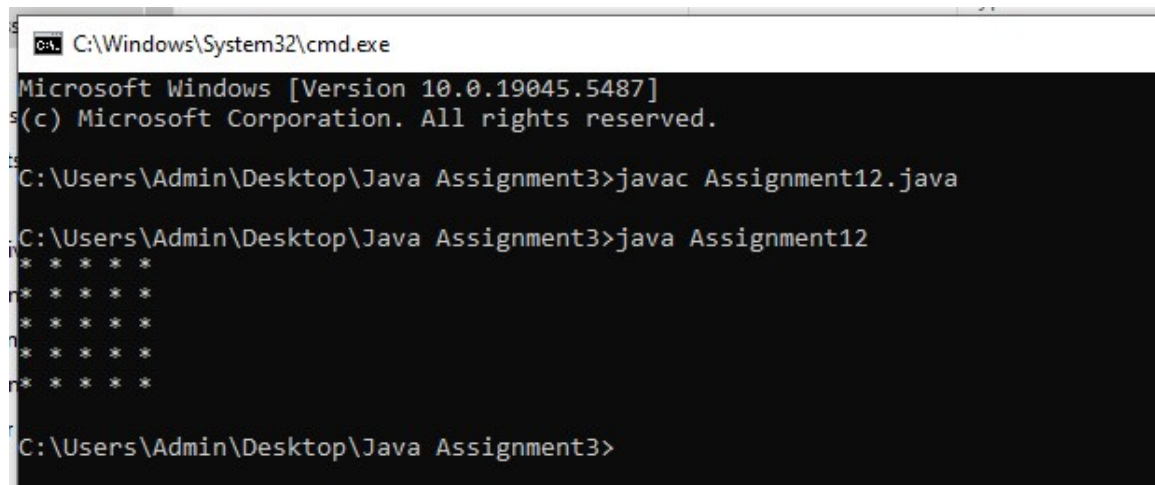
```
        }
```

```
        System.out.println();
```

```
    }
```

```
}
```

```
}
```



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment12.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment12
*****
*****
*****
*****
*****

C:\Users\Admin\Desktop\Java Assignment3>
```

```
/*
```

13. Write a program to print the following pattern:

1

2*2

3*3*3

4*4*4*4

5*5*5*5*5

5*5*5*5*5

4*4*4*4

3*3*3

2*2

1

```
*/
```

```
class Assignment13{
```

```
public static void main(String[] args){
```

```
    int n = 5;
```

```
    // upper part
```

```
    for(int i=1; i<=n; i++){
```

```
        for(int j=1; j<=i; j++){
```

```
            System.out.print(i);
```

```
            if(j<i){
```

```
                System.out.print("*");
```

```
            }
```

```
        }
```

```
        System.out.println();
```

```

    }
    // lower part
    for(int i=n; i>=1; i--){
        for(int j=1; j<=i; j++){
            System.out.print(i);
            if(j<i){
                System.out.print("*");
            }
        }
        System.out.println();
    }
}
}
}

```

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment13.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment13
1
12*2
123*3*3
1234*4*4*4
12345*5*5*5*5
12345*5*5*5*5
1234*4*4*4
123*3*3
12*2
1

C:\Users\Admin\Desktop\Java Assignment3>

```

/*14. Write a program to print the following pattern:

*

```

**
***
****
*****
*****
*****
*/
public class Assignment14 {
    public static void main(String[] args) {
        for (int i = 1; i <= 6; i++) {
            for (int j = 1; j <= i; j++) {
                if (i == 1) {
                    System.out.print("*");
                } else {
                    System.out.print("* ");
                }
            }
            System.out.println();
        }
    }
}

```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment14.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment14
*
* *
* * *
* * * *
* * * * *
* * * * *

C:\Users\Admin\Desktop\Java Assignment3>
```

/*15. Write a program to print the following pattern:

*

**

*/

```
public class Assignment15 {
    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment15.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment15
*
**
***
****
*****

C:\Users\Admin\Desktop\Java Assignment3>_
```

/*16. Write a program to print the following pattern:

```
*
**
***
****
*****
*****
*****
```

*/

```
public class Assignment16 {
    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++) {
            for (int j = 1; j <= 2 * i - 1; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment16.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment16
*
***
*****
*****
*****

C:\Users\Admin\Desktop\Java Assignment3>_
```

/*17. Write a program to print the following pattern:

**

*

*/

```
public class Assignment17 {
    public static void main(String[] args) {
        for (int i = 5; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment17.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment17
*****
*****
***
**
*

C:\Users\Admin\Desktop\Java Assignment3>_
```

/*18. Write a program to print the following pattern:

```
*
***
*****
*****
*****
***
*
*/
```

```
public class Assignment18 {
    public static void main(String[] args) {
        for (int i = 1; i <= 4; i++) {
            for (int j = 1; j <= 2 * i - 1; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
        for (int i = 3; i >= 1; i--) {
            for (int j = 1; j <= 2 * i - 1; j++) {
```



```

        System.out.print("*");
    }
    System.out.println();
}
}
}

```

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment18.java
C:\Users\Admin\Desktop\Java Assignment3>java Assignment18
*
***
*****
*****
*****
***
*
C:\Users\Admin\Desktop\Java Assignment3>

```

/*19. Write a program to print the following pattern:

```

1
1*2
1*2*3
1*2*3*4
1*2*3*4*5
*/

```

```

public class Assignment19 {
    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++) {
            for (int j = 1; j <= i; j++) {

```

```

        System.out.print(j);

        if (j < i) {
            System.out.print("*");
        }

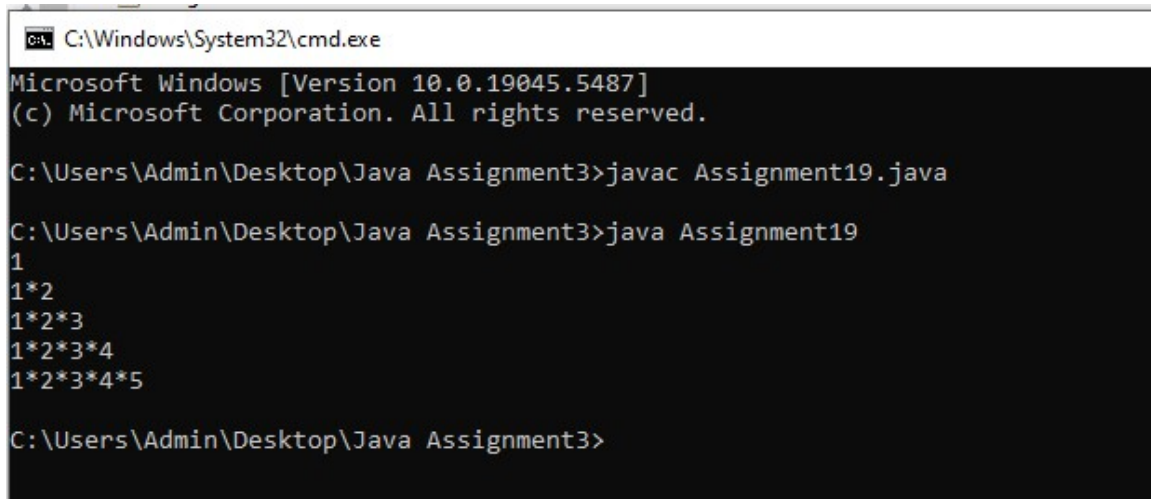
    }

    System.out.println();
}

}

}

```



```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment19.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment19
1
1*2
1*2*3
1*2*3*4
1*2*3*4*5

C:\Users\Admin\Desktop\Java Assignment3>

```

/*20. Write a program to print the following pattern:

```

5
5*4
5*4*3
5*4*3*2
5*4*3*2*1
*/

```

```

public class Assignment20 {
    public static void main(String[] args) {
        int num = 5;
    }
}

```

```

    for (int i = 1; i <= 5; i++) {
        for (int j = 1; j <= i; j++) {
            System.out.print(num);

            if (j < i) {
                System.out.print("*");
            }
        }
        num--;
        System.out.println();
    }
}

```

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment20.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment20
5
4*4
3*3*3
2*2*2*2
1*1*1*1*1

C:\Users\Admin\Desktop\Java Assignment3>

```

/*21. Write a program to print the following pattern:

```

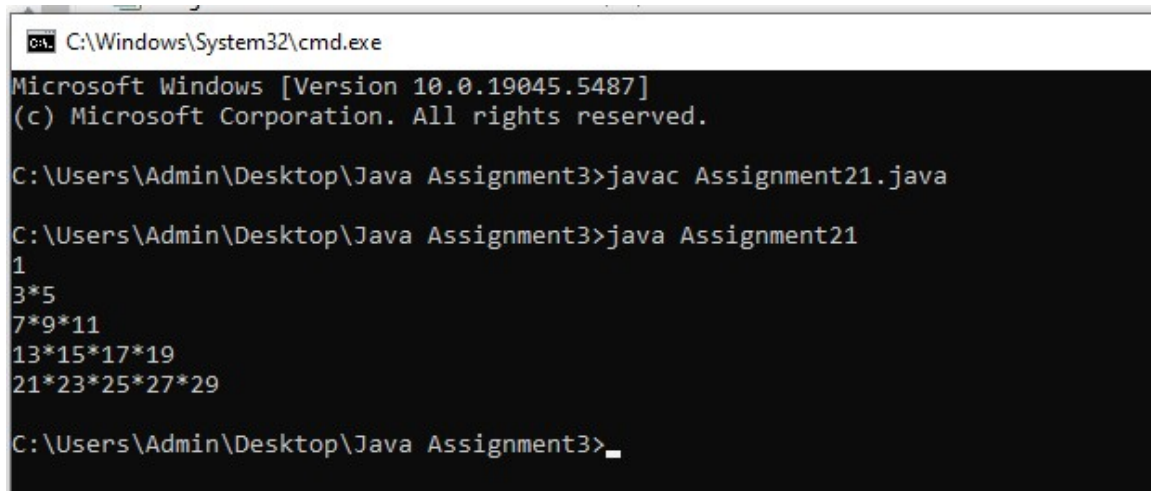
1
1*3
1*3*5
1*3*5*7
1*3*5*7*9

```

```

*/
public class Assignment21 {
    public static void main(String[] args) {
        int num = 1;
        for (int i = 1; i <= 5; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(num);
                if (j < i) {
                    System.out.print("*");
                }
                num += 2; // increment by 2 to get odd numbers
            }
            System.out.println();
        }
    }
}

```



```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment21.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment21
1
3*5
7*9*11
13*15*17*19
21*23*25*27*29

C:\Users\Admin\Desktop\Java Assignment3>_

```

/*22. Write a program to print the following pattern:

*

*/

```
public class Assignment22 {  
    public static void main(String[] args) {  
        for (int i = 9; i >= 1; i -= 2) {  
            for (int j = 1; j <= i; j++) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
        for (int i = 3; i <= 9; i += 2) {  
            for (int j = 1; j <= i; j++) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
    }  
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment22.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment22
*****
*****
*****
***
*
***
*****
*****
*****

C:\Users\Admin\Desktop\Java Assignment3>
```

/*23. Write a program to print the following pattern:

11111

22222

33333

44444

55555

*/

```
public class Assignment23 {
    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++) {
            for (int j = 1; j <= 5; j++) {
                System.out.print(i);
            }
            System.out.println();
        }
    }
}
```

```
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment23.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment23
11111
22222
33333
44444
55555

C:\Users\Admin\Desktop\Java Assignment3>_
```

/*24. Write a program to print the following pattern:

1

22

333

4444

55555

*/

```
public class Assignment24 {
    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(i);
            }
            System.out.println();
        }
    }
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment24.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment24
1
22
333
4444
55555

C:\Users\Admin\Desktop\Java Assignment3>_
```

/*25. Write a program to print the following pattern:

1

12

123

1234

12345

*/

```
public class Assignment25 {
    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(j);
            }
            System.out.println();
        }
    }
}
```



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment25.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment25
1
12
123
1234
12345

C:\Users\Admin\Desktop\Java Assignment3>
```

/*26. Write a program to print the following pattern:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

*/

```
public class Assignment26 {
    public static void main(String[] args) {
        for (int i = 1; i <= 15; i++) {
            System.out.print(i + " ");
        }
    }
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\Java Assignment3>javac Assignment26.java

C:\Users\Admin\Desktop\Java Assignment3>java Assignment26
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

C:\Users\Admin\Desktop\Java Assignment3>
```