Information Technology Concepts
Jobsheet 6

Name
Dicha Zelianivan Arkana

NIM
2241720002

Class
1i

Department
Information Technology

Study Program
D4 Informatics Engineering

Contents

1 Laboratory

1.1 Experiment 1: Calculate factorial values using iteration

a) Loop using **for**

1. Open a text editor. Create a new file, name it **FactorialFor.java**
2. Write the basic structure of Java programming language which contains the `main()` function.
3. Add the Scanner library
4. Make a **Scanner** declaration with the name `input`
5. Create multiple `int` type variables with names `number`, `factorial`, and `i`
6. Write down the syntax for entering the value from keyboard

```
System.out.print("Enter a number: ");
number = input.nextInt();
```

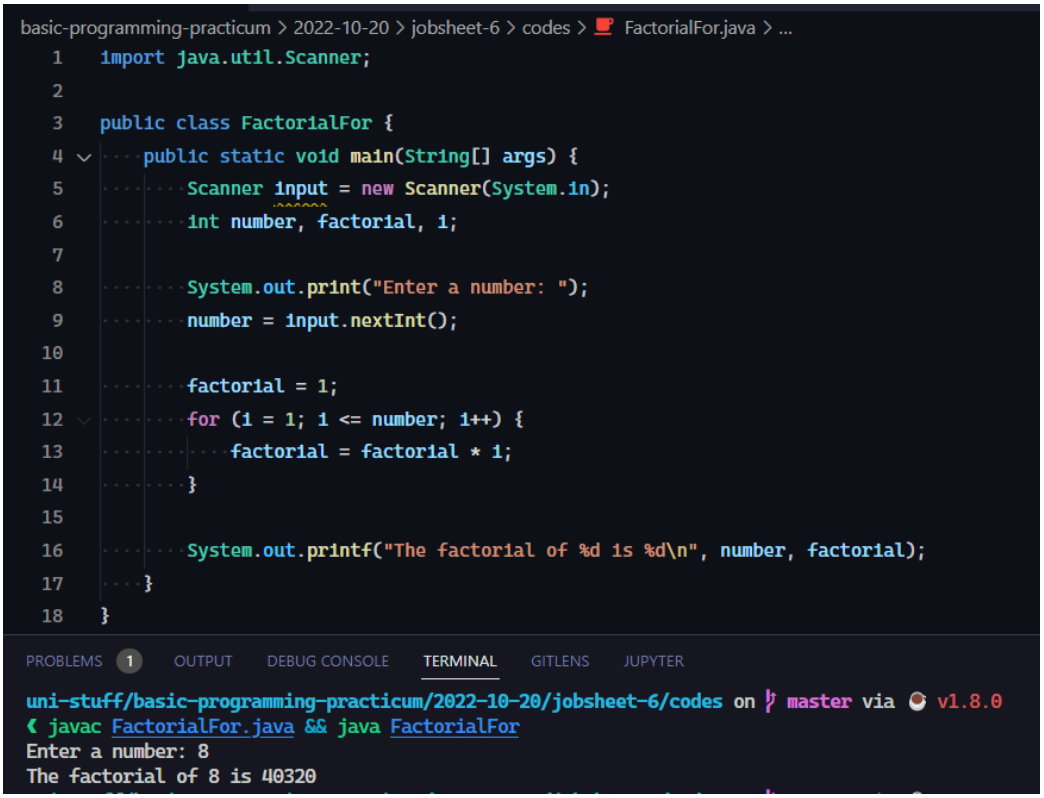
7. Create a for loop structure to calculate the factorial

```java
factorial = 1;
for (i = 1; i <= number; i++) {
    factorial = factorial * i;
}
```

8. Display factorial calculation result

```java
System.out.printf("The factorial of %d is %d\n", number, factorial);
```

9. Compile and run the program. Observe the results!


basic-programming-practicum > 2022-10-20 > jobsheet-6 > codes > FactorialFor.java > ...
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
GITLENS
JUPYTER

```java
import java.util.Scanner;

public class FactorialFor {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int number, factorial, i;

        System.out.print("Enter a number: ");
        number = input.nextInt();

        factorial = 1;
        for (i = 1; i <= number; i++) {
            factorial = factorial * i;
        }

        System.out.printf("The factorial of %d is %d\n", number, factorial);
    }
}
```

```
uni-stuff/basic-programming-practicum/2022-10-20/jobsheet-6/codes on master via v1.8.0
❮ javac FactorialFor.java && java FactorialFor
Enter a number: 8
The factorial of 8 is 40320
```

Figure 1: `FactorialFor.java` final code and output

b) Loop using **while**

1. Open a text editor. Create a new file, name it **FactorialWhile.java**
2. Write the basic structure of Java programming language which contains the `main()` function.
3. Add the Scanner library
4. Make a **Scanner** declaration with the name `input`

5. Create multiple `int` type variables with names `number`, `factorial`, and `i`
6. Write down the syntax for entering the value from keyboard

```
System.out.print("Enter a number: ");
number = input.nextInt();
```

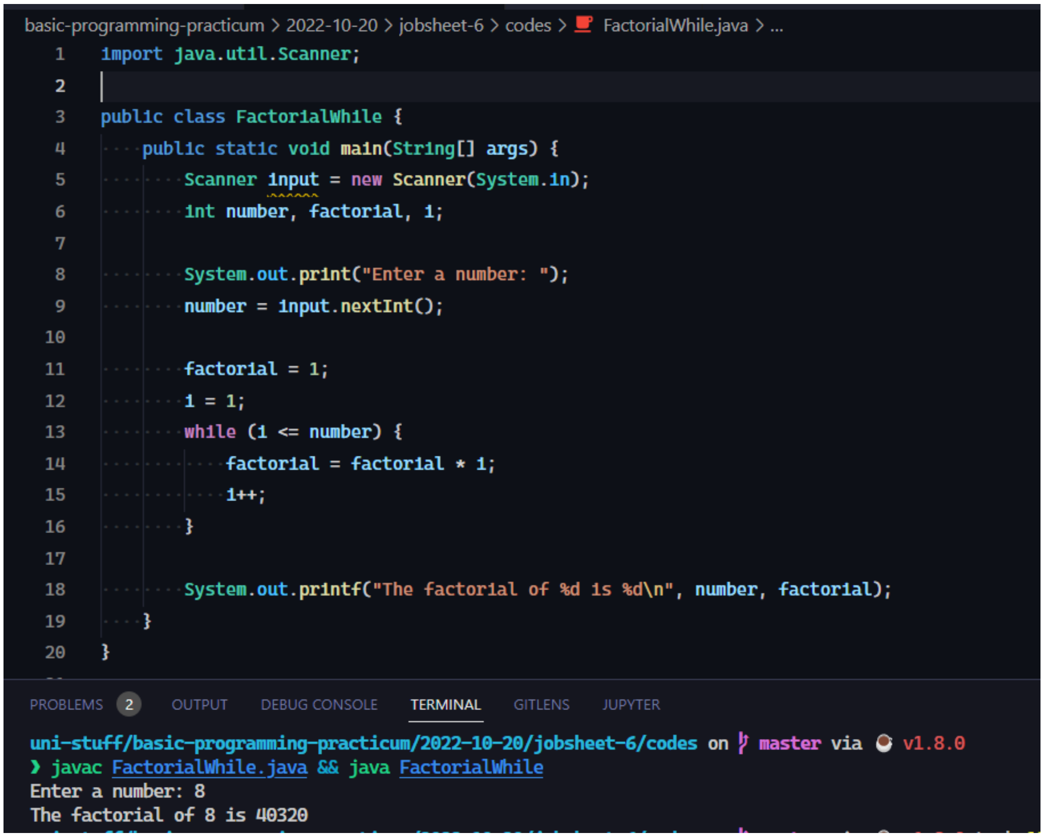
7. Create a while loop structure to calculate the factorial

```
factorial = 1;
i = 1;
while (i <= number) {
    factorial = factorial * i;
    i++;
}
```

8. Display factorial calculation result

```
System.out.printf("The factorial of %d is %d\n", number, factorial);
```

9. Compile and run the program. Observe the results!


basic-programming-practicum > 2022-10-20 > jobsheet-6 > codes > FactorialWhile.java > ...
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
GITLENS
JUPYTER

```
import java.util.Scanner;

public class FactorialWhile {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int number, factorial, i;

        System.out.print("Enter a number: ");
        number = input.nextInt();

        factorial = 1;
        i = 1;
        while (i <= number) {
            factorial = factorial * i;
            i++;
        }

        System.out.printf("The factorial of %d is %d\n", number, factorial);
    }
}
```

```
uni-stuff/basic-programming-practicum/2022-10-20/jobsheet-6/codes on  master via ☕ v1.8.0
❯ javac FactorialWhile.java && java FactorialWhile
Enter a number: 8
The factorial of 8 is 40320
```

Figure 2: `FactorialWhile.java` final code and output

c) Loop using **do-while**

1. Open a text editor. Create a new file, name it **FactorialDoWhile.java**
2. Write the basic structure of Java programming language which contains the `main()` function.
3. Add the Scanner library
4. Make a **Scanner** declaration with the name `input`
5. Create multiple `int` type variables with names `number`, `factorial`, and `i`
6. Write down the syntax for entering the value from keyboard

```
System.out.print("Enter a number: ");
number = input.nextInt();
```

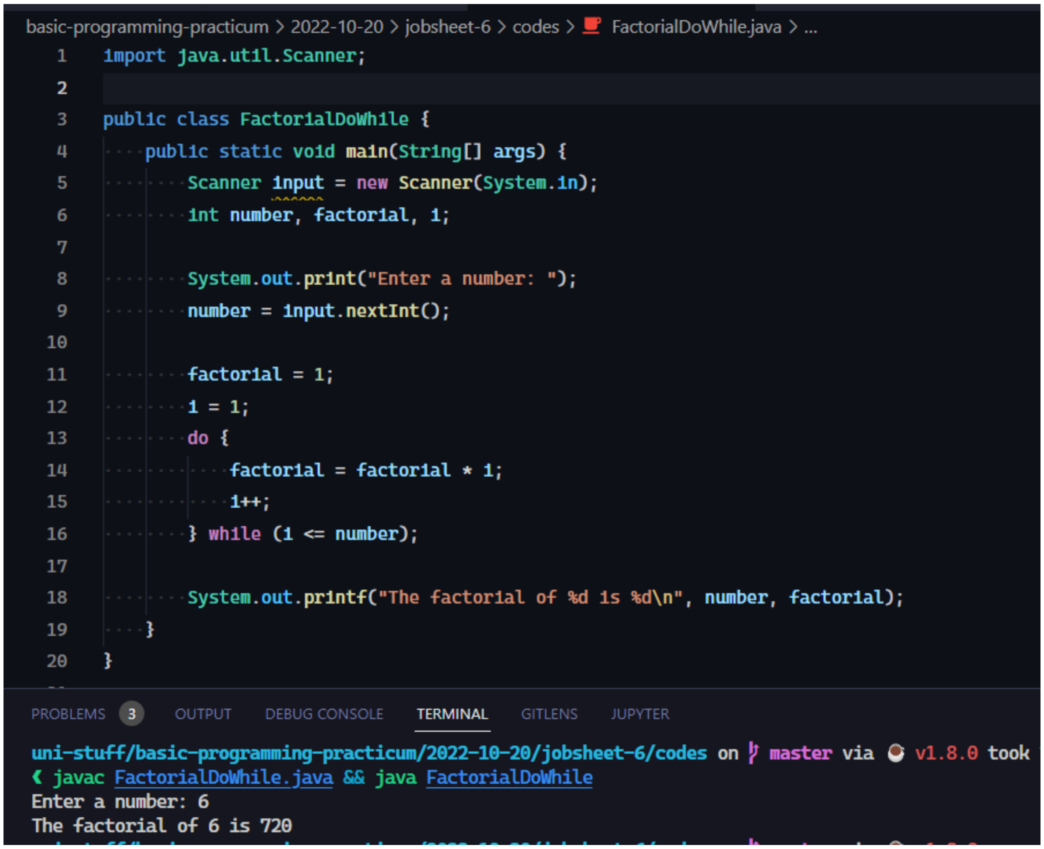
7. Create a do-while loop structure to calculate the factorial

```
factorial = 1;
i = 1;
do {
    factorial = factorial * i;
    i++;
} while (i <= number);
```

8. Display factorial calculation result

```
System.out.printf("The factorial of %d is %d\n", number, factorial);
```

9. Compile and run the program. Observe the results!



basic-programming-practicum > 2022-10-20 > jobsheet-6 > codes > FactorialDoWhile.java > ...
PROBLEMS 3
OUTPUT
DEBUG CONSOLE
TERMINAL
GITLENS
JUPYTER

```java
import java.util.Scanner;

public class FactorialDoWhile {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int number, factorial, i;

        System.out.print("Enter a number: ");
        number = input.nextInt();

        factorial = 1;
        i = 1;
        do {
            factorial = factorial * i;
            i++;
        } while (i <= number);

        System.out.printf("The factorial of %d is %d\n", number, factorial);
    }
}
```

```
uni-stuff/basic-programming-practicum/2022-10-20/jobsheet-6/codes on  master via  v1.8.0 took
❮ javac FactorialDoWhile.java && java FactorialDoWhile
Enter a number: 6
The factorial of 6 is 720
```

Figure 3: `FactorialDoWhile.java` final code and output

10. Match the results of the running programs that you have created according to the following display

```
Enter a number: 6
The factorial of 6 is 720
```

1.2 Experiment 2: Exit loop using break

1. Open a text editor. Create a new file, name it **LoopBreak.java**

2. Write the basic structure of Java programming language which contains the `main()` function.

3. Add the Scanner library

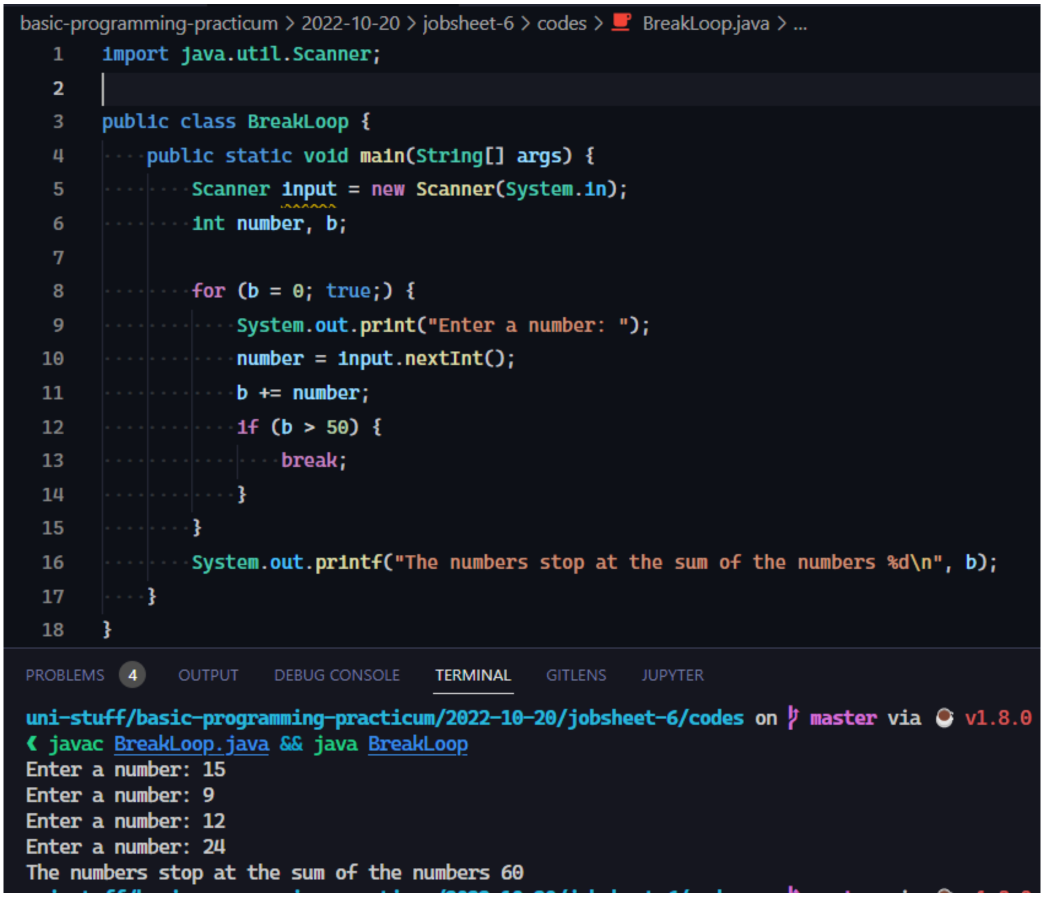
4. Make a **Scanner** declaration with the name `input`

5. Create multiple `int` type variables with names `number`, and `b`

6. Add the following code to enter the value from keyboard in 'for' loop structure. In 'for' loop there is also a condition to stop the process using the `break` statement.

```
for (b = 0; true;) {
    System.out.print("Enter a number: ");
    number = input.nextInt();
    b += number;
    if (b > 50) {
        break;
    }
}
System.out.printf("The numbers stop at the sum of the numbers %d\n", b);
```

7. Compile and run the program. Observe the results!


basic-programming-practicum > 2022-10-20 > jobsheet-6 > codes > BreakLoop.java > ...
PROBLEMS 4
OUTPUT
DEBUG CONSOLE
TERMINAL
GITLENS
JUPYTER

```java
import java.util.Scanner;

public class BreakLoop {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int number, b;

        for (b = 0; true;) {
            System.out.print("Enter a number: ");
            number = input.nextInt();
            b += number;
            if (b > 50) {
                break;
            }
        }
        System.out.printf("The numbers stop at the sum of the numbers %d\n", b);
    }
}
```

```
uni-stuff/basic-programming-practicum/2022-10-20/jobsheet-6/codes on  master via  v1.8.0
❮ javac BreakLoop.java && java BreakLoop
Enter a number: 15
Enter a number: 9
Enter a number: 12
Enter a number: 24
The numbers stop at the sum of the numbers 60
```

Figure 4: `BreakLoop.java` final code and output

8. Match the result of the running programs that you have created according to the following display

```
Enter a number: 15
Enter a number: 9
Enter a number: 12
Enter a number: 24
The numbers stop at the sum of the numbers 60
```

1.3 Experiment 3: Exit loop using continue

1. Open a text editor. Create a new file, name it **LoopBreak.java**

2. Write the basic structure of Java programming language which contains the `main()` function.

3. Add the Scanner library

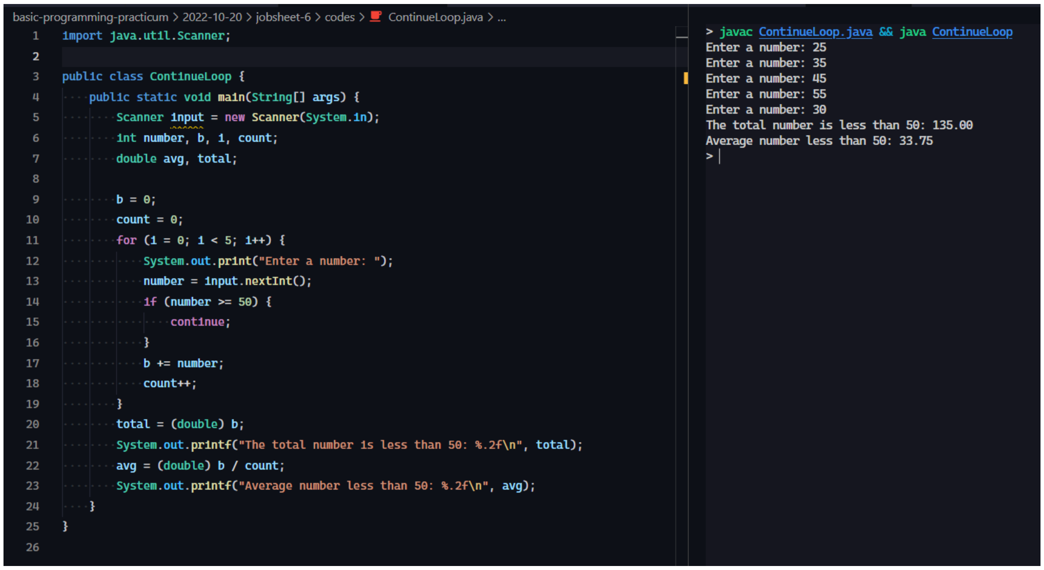
4. Make a **Scanner** declaration with the name `input`

5. Create multiple `int` type variables with names `number`, `b`, `i`, and `count`. Then also create two `double` type variables with names `avg` and `total`

6. Add the following code to enter the value from keyboard in ’for’ loop structure. In ’for’ loop there is also a condition to stop the process using the `continue` statement.

```
b = 0;
count = 0;
for (i = 0; i < 5; i++) {
    System.out.println("Enter a number: ");
    number = input.nextInt();
    if (number >= 50) {
        continue;
    }
    b += number;
    count++;
}
total = (double) b;
System.out.printf("The total number is less than 50: %.2f\n", total);
avg = (double) b / count;
System.out.printf("Average number less than 50: %.2f\n", avg);
```

7. Compile and run the program. Observe the results!



basic-programming-practicum > 2022-10-20 > jobsheet-6 > codes > ContinueLoop.java > ...

```java
import java.util.Scanner;

public class ContinueLoop {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int number, b, i, count;
        double avg, total;

        b = 0;
        count = 0;
        for (i = 0; i < 5; i++) {
            System.out.print("Enter a number: ");
            number = input.nextInt();
            if (number >= 50) {
                continue;
            }
            b += number;
            count++;
        }
        total = (double) b;
        System.out.printf("The total number is less than 50: %.2f\n", total);
        avg = (double) b / count;
        System.out.printf("Average number less than 50: %.2f\n", avg);
    }
}
```

```
> javac ContinueLoop.java && java ContinueLoop
Enter a number: 25
Enter a number: 35
Enter a number: 45
Enter a number: 55
Enter a number: 30
The total number is less than 50: 135.00
Average number less than 50: 33.75
> |
```

Figure 5: `ContinueLoop.java` final code and output

8. Match the result of the running programs that you have created according to the following display

```
Enter a number: 25
Enter a number: 35
Enter a number: 45
Enter a number: 55
Enter a number: 30
The total number is less than 50: 135.00
Average number less than 50: 33.75
```

2 Questions!

1. Explain the difference between Experiment 2 and Experiment 3!

   Experiment 2 uses the `break` keyword to stop the loop while Experiment 3 uses the `continue` keyword to stop the loop. The difference here is that the `break` keyword will 'break' out of the loop and it will *stop* the loop without executing any of the code below of this keyword. The `continue` keyword will *skip* the current iteration and 'continue' to the next iteration without executing the code below the keyword.

2. Suppose you are asked to create a java program that asks for input of an integer `n`. Then the program displays the character '*' on the screen **n times**. Which of the two pieces of the program below is better and safer? Why?

```java
/* for example: user input n has been stored in integer variable n */
int i = 0;

while (i < n) {
    System.out.print("*");
    i++;
}
```

```java
/* for example: user input n has been stored in integer variable n */
int i = 0;

while (i != n) {
    System.out.print("*");
    i++;
}
```

The first one (left) is better because it's more readable and its intent is more clear than the right one. We want the loop to run as long as `i` is less than `n` as opposed to `i` is not equal to `n`.

3. What is the output of the following three code snippets?

a.

```
int r = 1;
int i = 1;
int a = 2;
int n = 4;

while (i <= n) {
    r = r * a;
    i++;
}

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

The output of the code above is 6

b.

```
int n = 5;
boolean stop = false;

int i = 1;
while (!stop) {
    if (i >= n) {
        stop = true;
    } else {
        if (i % 2 == 1) {
            System.out.println("#");
        } else {
            System.out.println("*");
        }
    }
    i++;
}
```

The output of the code above is:

```
#
*
#
*
```

c.

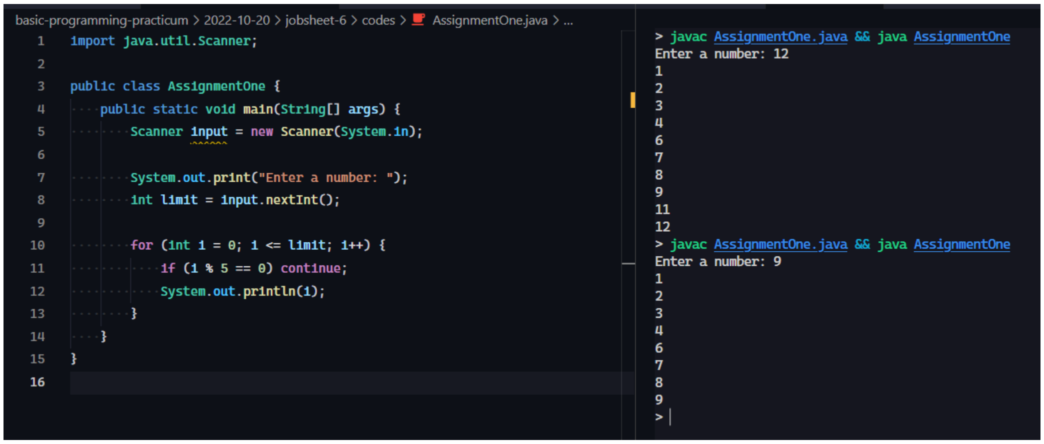
```
int n = 5;
long result = 1;
for (int i = 1; i <= n; i++) {
    result = result * i;
}
System.out.println(n + "!=" + result);
```

The output of the code above is: 5!=120

3 Assignment

1. Create a program to display numbers from 1 to the user input numbers sequentially and skip the multiples of 5 as shown below!

```
Enter a number: 12                Enter a number: 9
1                                 1
2                                 2
3                                 3
4                                 4
6                                 6
7                                 7
8                                 8
9                                 9
11
12
```



basic-programming-practicum > 2022-10-20 > jobsheet-6 > codes > AssignmentOne.java > ...

```java
import java.util.Scanner;

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

        System.out.print("Enter a number: ");
        int limit = input.nextInt();

        for (int i = 0; i <= limit; i++) {
            if (i % 5 == 0) continue;
            System.out.println(i);
        }
    }
}
```

```
> javac AssignmentOne.java && java AssignmentOne
Enter a number: 12
1
2
3
4
6
7
8
9
11
12
> javac AssignmentOne.java && java AssignmentOne
Enter a number: 9
1
2
3
4
6
7
8
9
> |
```

Figure 6: `AssignmentOne.java` final code and output

2. Create a program using the Java programming language that requests input of an integer N (N>0) from user. The program then displays the sum of **the first N positive even numbers (even numbers $\geq$ 0)**.

   Example:

   - If the user enters N = 10, the program will count the number of positive numbers in the range 1-10 then display the sum of positive numbers between 1-10, namely:

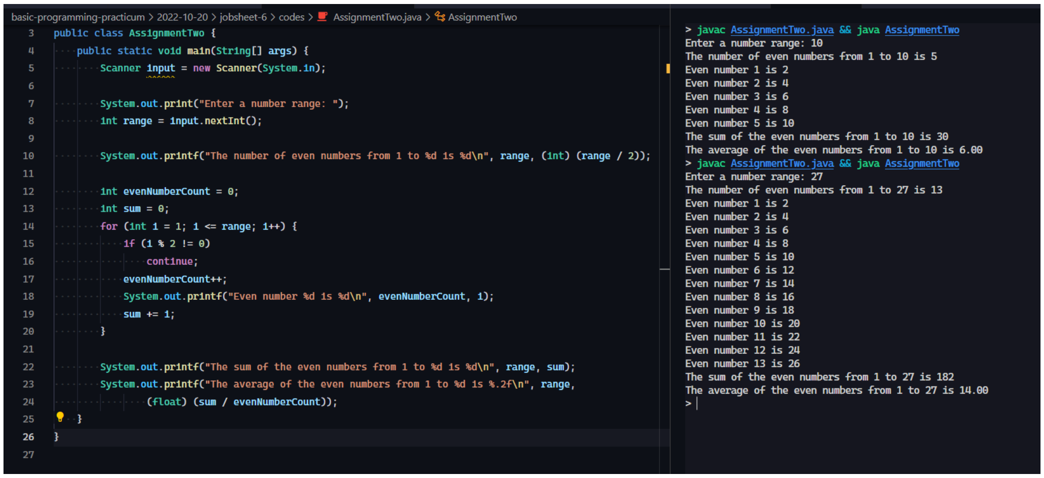
     $$0 + 2 + 4 + 6 + 8 + 10 = 30$$

     After that the program will display the average of the positive numbers that were added earlier.

- Example of program output

```
Enter a number range: 10
The number of even numbers from 1 to 10 is 5
Even number 1 is 2
Even number 2 is 4
Even number 3 is 6
Even number 4 is 8
Even number 5 is 10
The sum of the even numbers from 1 to 10 is 30
The average of the even numbers from 1 to 10 is 6.00
```

You can design your own for the program display


basic-programming-practicum > 2022-10-20 > jobsheet-6 > codes > AssignmentTwo.java > AssignmentTwo

```java
public class AssignmentTwo {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a number range: ");
        int range = input.nextInt();

        System.out.printf("The number of even numbers from 1 to %d is %d\n", range, (int) (range / 2));

        int evenNumberCount = 0;
        int sum = 0;
        for (int i = 1; i <= range; i++) {
            if (i % 2 != 0)
                continue;
            evenNumberCount++;
            System.out.printf("Even number %d is %d\n", evenNumberCount, i);
            sum += i;
        }

        System.out.printf("The sum of the even numbers from 1 to %d is %d\n", range, sum);
        System.out.printf("The average of the even numbers from 1 to %d is %.2f\n", range,
                (float) (sum / evenNumberCount));
    }
}
```

```
> javac AssignmentTwo.java && java AssignmentTwo
Enter a number range: 10
The number of even numbers from 1 to 10 is 5
Even number 1 is 2
Even number 2 is 4
Even number 3 is 6
Even number 4 is 8
Even number 5 is 10
The sum of the even numbers from 1 to 10 is 30
The average of the even numbers from 1 to 10 is 6.00
> javac AssignmentTwo.java && java AssignmentTwo
Enter a number range: 27
The number of even numbers from 1 to 27 is 13
Even number 1 is 2
Even number 2 is 4
Even number 3 is 6
Even number 4 is 8
Even number 5 is 10
Even number 6 is 12
Even number 7 is 14
Even number 8 is 16
Even number 9 is 18
Even number 10 is 20
Even number 11 is 22
Even number 12 is 24
Even number 13 is 26
The sum of the even numbers from 1 to 27 is 182
The average of the even numbers from 1 to 27 is 14.00
> 
```

Figure 7: `AssignmentTwo.java` final code and output