Basic Programming Practicum
Looping 2

Name
Dicha Zelianivan Arkana

NIM
2241720002

Class
1i

Department
Information Technology

Study Program
D4 Informatics Engineering

1 Laboratory
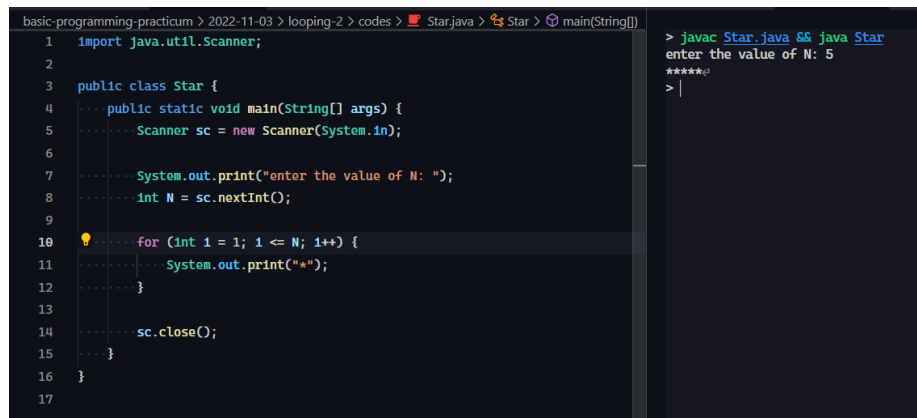
1.1 Experiment 1: Loop Review

1. Experiment 1 was aimed at reviewing the loop that had been studied in the previous week. In Experiment 1, a program will be made to make a view * N times sideways.

2. Create a new Class, name it **Star**

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

4. Add the Scanner library

5. Make a **Scanner** declaration with the name `sc`

6. Add the following code to receive input from keyboard as the value to be stored in the variable N.

```java
System.out.print("Enter the value of N: ");
int N = sc.nextInt();
```

7. Add a for loop structure to display the * symbol according to the number specified via input

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

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

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

```
Enter the value of N: 5
*****
```

Questions!

1. If in **for** loop, the initialization `i = 1` is changed to `i = 0`, what is the result? How can it be like that?

   If we change the initialisation to 0, then we'll get N+1 amount of stars because our condition is `i <= N` which means it will iterate from 0 until N. For example, if N is 5, it will go through 0, 1, 2, 3, 4, 5, which is 6 times.

2. If in **for** loop, condition `i <= N` is changed to `i > N`, what is the result? How can it be like that?

   It will print the stars indefinitely if we set the N variable to 0 because `i` will be greater than 0 and it will not stop because it doesn't have a stop condition.

3. If in **for** loop, the condition for step `i++` is changed to `i--` what is the result? How can it be like that?

   It will print the stars indefinitely if we set the N variable to greater than equal to 1 because the `i` comparison will always be true when it is smaller than or equal to N.

1.2 Experiment 2: Square star

1. Experiment 2 is used to create a display * in the form of a square, with sides of a number of N. When observed further, this problem is actually similar to Experiment 1. In Experiment 1, for example the input of N is 5, then the resulting output is `*****` (we can think of it as an inner loop showing 5 stars `*****`). For experiment 2, doesn't the result of the Experiment 1 just need to be repeated N times? (by adding an outer loop to repeat the inner loop process N times)

2. Create a new class, name it **Square**

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

4. Add the same program code as the contents of the `main()` function in Experiment 1

```java
Scanner sc = new Scanner(System.in);
System.out.print("Enter the value of N: ");
int N = sc.nextInt();
for (int i = 1; i <= N; i++) {
    System.out.print("*");
}
```

5. Run the program. Make sure the results given are the same as in Experiment 1

6. Pay attention to the iterative syntax used to print * N times sideways. In step 4, we make for loop structure (red box) as an **inner loop**

7. Furthermore, the inner loop needs to be repeated N times in order to display the * symbol to form a square. Thus, is is necessary to add an outer loop.

```java
for (int iOuter = 1; iOuter <= N; iOuter++) {
    for (int i = 1; i <= N; i++) {
        System.out.print("*");
    }
    System.out.println("");
}
```

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

9. Match the results of the running program that you have created according to the following display

```
Enter the value of N: 5
*****
*****
*****
*****
*****
```

Questions!

1. Pay attention to the outer loop. If in **for** syntax, the initialization `iOuter = 1` is changed to `iOuter = 0`, what is the result? How it can be like that?

   It will print the rows N+1 number of times because the comparison used is `<=`. For example, if the N is 5 then it will go through 0, 1, 2, 3, 4, 5, which is 6 times.

2. Return the program to normal with initialization `iOuter = 1`. Then pay attention to the inner loop. If in **for** syntax, the initialization `i = 1` is changed to `i = 0`, what is the result? How can it be like that?

   Each row will have an extra star because, like the first question, it will print N+1 times because the condition used is `<=`.

3. What is the difference between outer loop and inner loop?

   The inner loop is used to print the stars on each row while the outer loop is used to repeat the process to print the star row.

4. Why is is necessary to add the syntax `System.out.println();` under inner loop? What will happen if the syntax is omitted?

   It will not print a square, but instead it will print a continuous line of star because there is no newline for each rows.
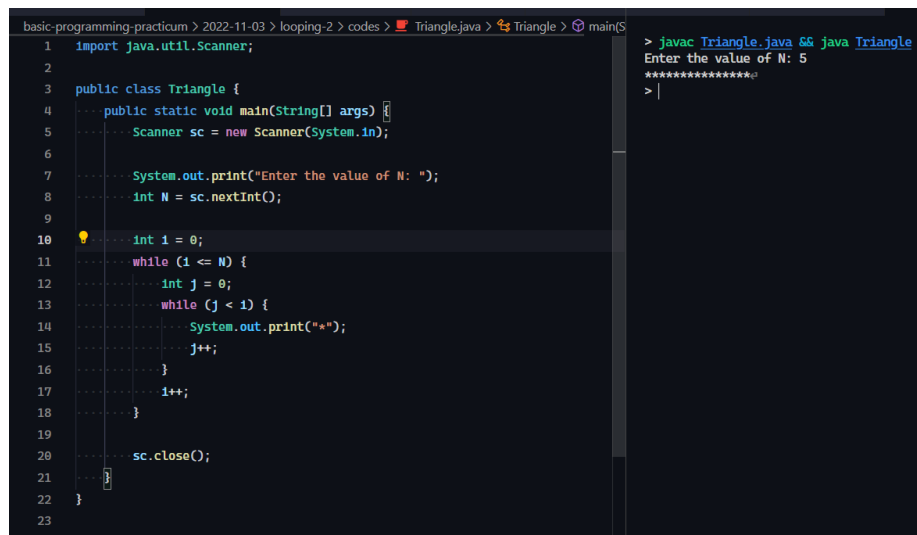
1.3   Experiment 3: Triangle Star

1. Experiment 3 is used to create a display * in the form of a right triangle with a height of N

2. Create a new class, name it **Triangle**

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

4. Add the Scanner library

5. Make a **Scanner** declaration with the name `sc`

6. Add the following code to receieve input from keyboard as the value to be stored in the variable N

   ```
   System.out.print("Enter the value of N: ");
   int N = sc.nextInt();
   ```

7. Add a while loop structure to display the * symbol according to the number specified via input

```java
    int i = 0;
    while (i <= N) {
        int j = 0;
        while (j < i) {
            System.out.print("*");
            j++;
        }
        i++;
    }
```

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

Questions!

1. Look at the results, is the output generated with a value of N = 5 in accordance with the following display?

   ```
   *
   **
   ***
   ****
   *****
   ```
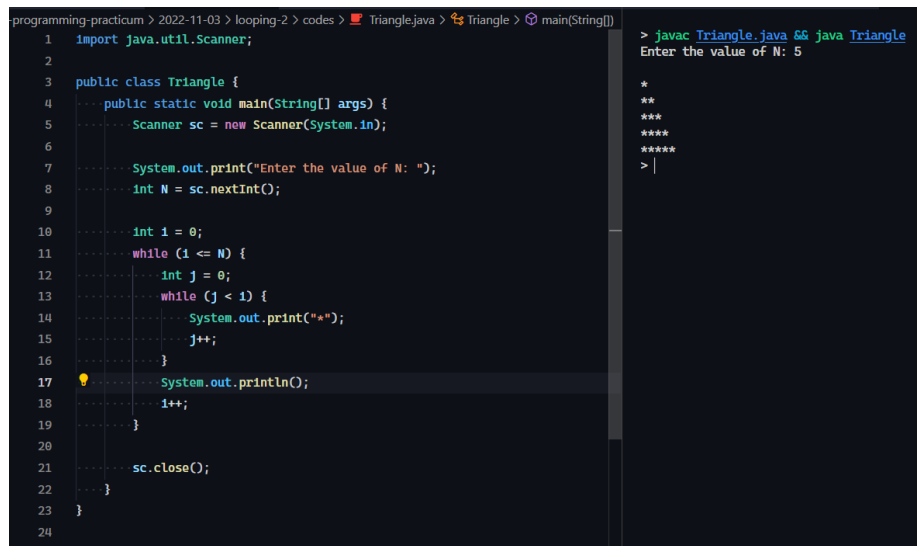
   No. It instead look like this

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

2. If not, which parts should be improved or added? Describe any parts that need to be improved or added!

   There should be an extra `System.out.println()` statement after the inner loop to break each row into its own line.
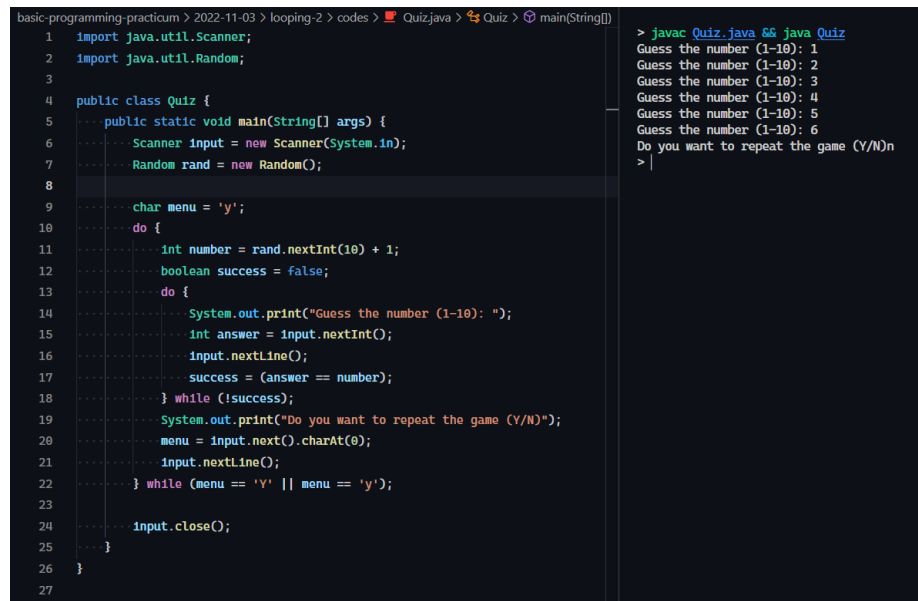

1.4 Experiment 4: Guess the Number Quiz

1. Experiment 4 is used to create a quiz to guess a random computer set number

2. Create a new class name, name it **Quiz**

3. Add Scanner and Random libraries outside the class

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

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

5. Make a `Scanner` declaration with the name `input` and `Random` declaration with the name `rand`

6. Add the following code to create a do-while loop structure that is used to make a game of guessing numbers quiz. In inner loop, the loop is used to ask the user to enter a number as long as the number entered does not match the number determined by the computer randomly. While the outer loop is used to repeat the game by choosing a new random number.

```java
Scanner input = new Scanner (System.in);
Random rand = new Random();
char menu = 'y';
do
{
    int number = rand.nextInt(10) + 1;
    boolean success = false;
    do
    {
        System.out.print("Guess the number (1-10): ");
        int answer = input.nextInt();
        input.nextLine();
        success = (answer == number);
    }
    while (!success);
    System.out.print("Do you want to repeat the game (Y/N)");
    menu = input.next().charAt(0);
    input.nextLine();
}
while (menu == 'Y' || menu == 'y');
```

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

Questions!

1. Explain the program flow in Experiment 4!

   - First, it will generate a random number ranging from 1 to 10 using `rand.nextInt(10) + 1`

   - It will start a loop what will prompt a user for a number between 1 and 10

   - Then, it will check if the user's answer matches with the generated random number

   - If it matches, it will set the `success` variable which in turn will break the `do-while` loop

   - It will then ask if the user want to play again or not

   - If they type 'y' or 'Y', the program will restart, if they don't, the program will stop because the condition in the while block is not satisfied.

2. What must be done to discontinue (not repeat) the game?

   When prompted to insert Y or N, we can insert anything except `y` or `Y` to stop the game.

3. Modify the program above, so that it can display information about: input the guess value entered by the user, whether it is smaller or greater than the answer (number) randomly determined by the computer!
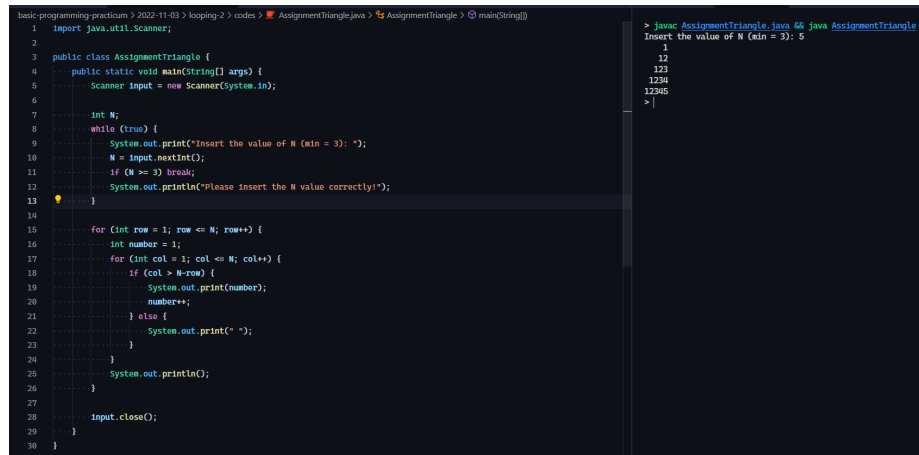
```java
import java.util.Scanner;
import java.util.Random;

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

        char menu = 'y';
        do {
            int number = rand.nextInt(10) + 1;
            boolean success = false;
            do {
                System.out.print("Guess the number (1-10): ");
                int answer = input.nextInt();
                input.nextLine();
                success = (answer == number);
                if (!success) {
                    System.out.printf("The number you inserted is too " + (answer > number ? "big" : "small") + "\n");
                }
            } while (!success);
            System.out.print("Do you want to repeat the game (Y/N)");
            menu = input.next().charAt(0);
            input.nextLine();
        } while (menu == 'Y' || menu == 'y');

        input.close();
    }
}
```

```
> javac Quiz.java && java Quiz
Guess the number (1-10): 5
The number you inserted is too big
Guess the number (1-10): 1
The number you inserted is too small
Guess the number (1-10): 3
The number you inserted is too big
Guess the number (1-10): 2
Do you want to repeat the game (Y/N)
```
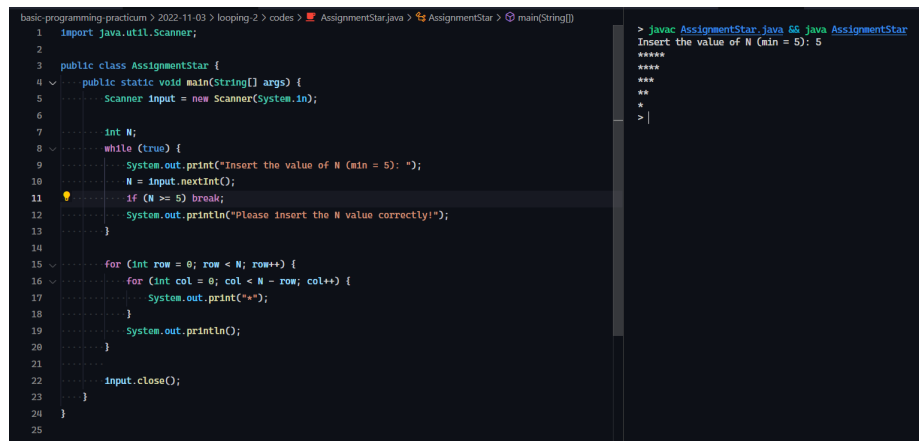
2 Assignments

1. Create a program to print a numeric triangle display as below based on the N input (minimum N value is 3). Example N = 5

```java
import java.util.Scanner;

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

        int N;
        while (true) {
            System.out.print("Insert the value of N (min = 3): ");
            N = input.nextInt();
            if (N >= 3) break;
            System.out.println("Please insert the N value correctly!");
        }

        for (int row = 1; row <= N; row++) {
            int number = 1;
            for (int col = 1; col <= N; col++) {
                if (col > N-row) {
                    System.out.print(number);
                    number++;
                } else {
                    System.out.print(" ");
                }
            }
            System.out.println();
        }

        input.close();
    }
}
```

```
> javac AssignmentTriangle.java && java AssignmentTriangle
Insert the value of N (min = 3): 5
    1
   12
  123
 1234
12345
> |
```

2. Create a program to print the star triangle view shown below based on the N input (minimum N value is 5). Example N = 7

```java
import java.util.Scanner;

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

        int N;
        while (true) {
            System.out.print("Insert the value of N (min = 5): ");
            N = input.nextInt();
            if (N >= 5) break;
            System.out.println("Please insert the N value correctly!");
        }

        for (int row = 0; row < N; row++) {
            for (int col = 0; col < N - row; col++) {
                System.out.print("*");
            }
            System.out.println();
        }

        input.close();
    }
}
```

```
> javac AssignmentStar.java && java AssignmentStar
Insert the value of N (min = 5): 5
*****
****
***
**
*
> |
```

3. Create a program to print a square numeric display like the one below based on N input (minimum N value is 3). Example N = 3 and N = 5

4. Create a program to print a square numeric display like the one below based on N input (minimum N value is 5). Example N = 5