Assignment 3

Snippet 1

1. **Error:** The loop runs infinitely because `i--` decreases the value of `i`, the condition aways remain true .
2. **Correction:** Change `i--` to `i++` to increment `i` correctly.

```
for (int i = 0; i < 10; i++) {
    System.out.println(i);
}
```

Snippet 2

1. **Error:** The condition `count = 0` assigns `0` to `count` instead of comparing it, making the loop always false. (Type mismatch error)
2. **Correction:** Use `count == 0` to compare the value of `count`.

```
while (count == 0) {
    System.out.println(count);
    count--;
}
```

Snippet 3

1. **Error:** The loop executes for infinity .
2. **Correction:** Adjust the condition to ensure multiple iterations.

```
do {
    System.out.println(num);
    num++;
} while (num < 5);
```

Snippet 4

1. **Error:** The loop runs 10 times, but the task expected only numbers 1 to 9.

2. **Correction:** Change the condition to `i < 10` to stop at 9.

```
for (int i = 1; i < 10; i++) {
   System.out.println(i);
}
```

Snippet 5

1. **Error:** The loop does not print the desired output because the loop will never terminate , as the i value will always. be greater than 10.
2. **Correction:** Change `i++` to `i--` to decrement `i`.

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

Snippet 6

1. **Error:** "Done" prints only once because it's outside the loop body.
2. **Correction:** Use curly braces `{}` to include both statements inside the loop.

```
for (int i = 0; i < 5; i++) {
   System.out.println(i);
   System.out.println("Done");
}
```

Snippet 7

1. **Error:** The variable `count` is not initialized, leading to a compilation error.
2. **Correction:** Initialize `count` before the loop.

```
int count = 0;
while (count < 10) {
   System.out.println(count);
   count++;
}
```

Snippet 8

1. **Error:** The loop prints unexpected numbers because `num--` decreases `num`, making the condition `num > 0` false .
2. **Correction:** increment `num` to print numbers 1 to 5.

```
int num = 1;
do {
   System.out.println(num);
   num++;
} while (num <= 5);
```

Snippet 9

1. **Error:** There is no error in the code .

Snippet 10

1. **Error:** The loop runs indefinitely because `num = 10` is an assignment, not a comparison and while needs condition to be checked .
2. **Correction:** Use `num == 10` to compare values.

```
while (num == 10) {
   System.out.println(num);
   num--;
}
```

Snippet 11

1. **Error:** There is no error in the code .

Snippet 12

1. **Error:** The variable `x` is out of scope outside the loop, causing a compilation error.
2. **Correction:** Declare `x` outside the loop if it needs to be used later.

```
int x = 0;
for (int i = 0; i < 5; i++) {
   x = i * 2;
}
```

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

SECTION 2: Guess the Output

- 1 1 1 2 2 1 2 2 3 1 3 2
- 11
- 0 1 2 3
- 12345
- 3
- 8
- 49
- -4

SECTION 3: Lamborghini Exercise:

1. Program to calculate the sum of the first 50 natural numbers:

```
public class SumNaturalNumbers {
    public static void main(String[] args) {
        int sum = 0;
        for (int i = 1; i <= 50; i++) {
            sum += i;
             }
      M      System.out.println("Sum of the first 50 natural numbers is: " + sum);
    }
}
```

2. Program to compute the factorial of the number 10:

```
public class Factorial {
    public static void main(String[] args) {
        int num = 10;
    long factorial = 1;
        for (int i = 1; i <= num; i++) {

  factorial *= i;
        }
        System.out.println("Factorial of 10 is: " + factorial);
    }
}
```

```
}
```

3. Program to print all multiples of 7 between 1 and 100:

```
public class MultiplesOfSeven {
       public static void main(String[] args) {
     for (int i = 7; i <= 100; i += 7) {
            System.out.println(i);
        }
    }}
```

4. Program to reverse the digits of the number 1234:

```
public class ReverseNumber {
          public static void main(String[] args) {
        int num = 1234;
           int reversed = 0;
        while (num != 0) {
               int digit = num % 10;
            reversed = reversed * 10 + digit;
              num /= 10;
        }
           System.out.println("Reversed number is: " + reversed);
    }
}
```

5. Program to print the Fibonacci sequence up to the number 21:

```
public class Fibonacci {
    public static void main(String[] args) {
        int a = 0, b = 1;
   System.out.print(a + " " + b);
        int next = a + b;
        while (next <= 21)
         System.out.print(" " + next);
            a = b;
            b = next;
            next = a + b;
        }
    }
}
```

6. Program to find and print the first 5 prime numbers:

```
public class FirstFivePrimes {
    public static void main(String[] args) {
        int count = 0;
      int num = 2;
        while (count < 5) {
            if (isPrime(num)) {
             System.out.println(num);
                count++;
            }
            num++;
        }
    }

    public static boolean isPrime(int n) {
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

7. Program to calculate the sum of the digits of the number 9876:

```
public class SumOfDigits {    public static void main(String[] args) {
        int num = 9876;
        int sum = 0;
        while (num != 0) {
           sum += num % 10;
            num /= 10;
        }
           System.out.println("Sum of the digits is: " + sum);
       }
}
```

8. Program to count down from 10 to 0, printing each number:

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


        }
    }
}
```

9. Program to find and print the largest digit in the number 4825:

```
public class LargestDigit {
    public static void main(String[] args) {
            int num = 4825;
        int maxDigit = 0;
        while (num != 0) {
            int digit = num % 10;
                 if (digit > maxDigit) {
                maxDigit = digit;
            }
            num /= 10;
        }
        System.out.println("Largest digit is: " + maxDigit);
    }
}
```

10. Program to print all even numbers between 1 and 50:

```
   public class EvenNumbers {
    public static void main(String[] args) {
      for (int i = 2; i <= 50; i += 2) {
            System.out.println(i);
        }
    }
}
```

11. Java program to demonstrate the use of both pre-increment and post-decrement operators in a single expression:

```
public class IncrementDecrement {
    public static void main(String[] args) {
                  int a = 5;
        int result = ++a - a--;
             System.out.println("Result of the expression is: " + result);
    }
}
```

12. Program to draw the pattern:

```
public class StarPattern {
            public static void main(String[] args) {
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 5; j++) {
                  System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

14. Program to print the pattern:

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

```
public class Pattern14 {
    public static void main(String[] args) {
        for (int i = 1; i <= 7; i++) {
                  for (int j = 1; j <= (2 * i - 1); j++) {
```

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

15. Program to print the pattern:

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

```
public class Pattern15 {
    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();
        }
    }
}
```

16. Program to print the pattern:

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

```
public class Pattern16 {
    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();
        }
  }
}
```

17. Program to print the pattern:

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

```
public class Pattern17 {
    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()       }
    }
}
```

23. Program to print the pattern:

```
11111
22222
33333
44444
55555
```

```
public class Pattern23 {
    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();
        }
    }
}
```

24. Program to print the pattern:

```
1
22
333
4444
55555
```

```
public class Pattern24 {
    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();
        }
    }
}
```

25. Program to print the pattern:

```
1
12
```

```
123
1234
12345
```

```
public class Pattern25 {
    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();
        }
    }
}
```

26. Program to print the pattern:

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

```
public class Pattern26 {
    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 + " ");
                num++;
            }
            System.out.println();
        }}}
```