Assignment - 3

Snippet 1:

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

// Error to investigate: Why does this loop run infinitely? How should the loop control variable be adjusted?
 The loop condition is i < 10, and i is being decremented (moving further away from 10), the condition will always remain true. As a result, the loop never exits and runs infinitely.

The loop control variable be adjusted by incrementing i as i++ in the for loop.

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

Snippet 2:

```
public class IncorrectWhileCondition {
    public static void main(String[] args) {
        int count = 5;
        while (count = 0) {
            System.out.println(count);
            count--;
        }
    }
}
```

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

The loop does not executed as expected due to condition in while loop, In the condition while (count = 0), the single equals sign = is an assignment operator, not a comparison operator. This statement assigns the value 0 to the variable count rather than comparing count with 0.
The result of the assignment count = 0 is 0, which is considered false for while loop. As a result, the loop never executes because the condition is always false.

Snippet 3:

```
    public class DoWhileIncorrectCondition {
        public static void main(String[] args) {
            int num = 0;
            do {
                System.out.println(num);
                    num++;
            } while (num > 0);
    }
}
```

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

The loop executes infinitely. This is due to the condition in do-while loop. If we put condition n<0 then it will give 0 output and stop.

Snippet 4:

```
public class OffByOneErrorForLoop {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
// Expected: 10 iterations with numbers 1 to 10
// Actual: Prints numbers 1 to 10, but the task expected only
    }
}
```

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

The loop should run upto 9 to achieve expected output i.e 1 to 9.

Corrected code -

```
public class Main {
    public static void main(String[] args) {
        for (int i = 1; i < 10; i++) {
            System.out.println(i);
        }
    }
}
```

Snippet 5:

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

// 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 for loop runs infinitely due to given condition. Initialization is right if but update statement is wrong if you want numbers from 10 to 0 in decrement order.

Snippet 6:

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

// 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 reason "Done" prints only once, outside the loop, is due to how Java interprets the loop's body when braces {} are not used. When a for loop (or any loop) is written without braces {}, only the first statement after the loop is considered part of the loop's body. Here, only System.out.println(i); is inside the loop, so it gets executed on each iteration. The System.out.println("Done"); statement is not part of the loop and is executed only once after the loop has completed.
To include both System.out.println(i); and System.out.println("Done"); inside the loop, you need to use braces {} to define the loop body:

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

Snippet 7:

```
public class UninitializedWhileLoop {
    public static void main(String[] args) {
        int count;
        while (count < 10) {
            System.out.println(count);
            count++;
        }
    }
}
```

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

```
Main.java:4: error: variable count might not have been init
ialized
        while (count < 10) {
               ^
1 error
```

The variable count is not initialized, Hence compilation error. To initialize the loop variable properly, we need to initialize count variable.