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?

Answer:

this loop run infinitely as condition given in the code is improper instead of i--, it should be 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?

Ans:

The condition of the while loop is incorrect to execute this code me should give condition as count<=0.

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?
Ans:

While condition is incorrect as num is already grater than 1 so it it will be in infinite loop instead of 0 we can give a finite number

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 1 to 9
	 }
}
```

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

Ans:

The condition i <= 10 causes the loop to iterate 10 times, printing numbers from 1 to 10.

However, the expected output suggests that the loop should print only numbers 1 to 9.

Adjust the loop condition from i <= 10 to i < 10, ensuring that it stops at i = 9.

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?

Ans:

In for loop initialization is not proper it should start from 0 to have the expected output

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?

By using {} we can have expected outcome

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?

Ans:

It gives compilation error because count is not initialize we need to initialize to execute the code properly

Snippet 8:

```
public class OffByOneDoWhileLoop {
    public static void main(String[] args) {
        int num = 1;
        do {
            System.out.println(num);
            num--;
        } while (num > 0);
    }
}
```

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

Ans:

As do loop done first done without any condition num is decremented do it become 0 which is not allowed in while loop as per condition num should be < 6 or <= 5 and befor loop end instead of decrement we need to increment num

Snippet 9:

```
public class InfiniteForLoopUpdate {
```

```
    public static void main(String[] args) {
        for (int i = 0; i < 5; i += 2) {
            System.out.println(i);
        }
    }
}
```

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

Ans:

Snippet 10:

```
public class IncorrectWhileLoopControl {
    public static void main(String[] args) {
        int num = 10;
        while (num = 10) {
            System.out.println(num);
            num--;
        }
    }
}
```

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

Snippet 11:

```
public class IncorrectLoopUpdate {
    public static void main(String[] args) {
        int i = 0;
        while (i < 5) {
            System.out.println(i);
            i += 2; // Error: This may cause unexpected results in output
```

```
                }
        }
}
```

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

Ans:

output will be 0,2,4

Instead of i= +2 we can have increment i++

Snippet 12:

```
public class LoopVariableScope {
        public static void main(String[] args) {
                for (int i = 0; i < 5; i++) {
                        int x = i * 2;
                }
                System.out.println(x); // Error: 'x' is not accessible here
        }
}
```

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