## Snippet 01

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

**Error:** This loop runs infinitely because `i` is decremented (`i--`) instead of incremented. The condition `i < 10` will always be true.

**Correction:** Change `i--` to `i++`.

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

---

## Snippet 02

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

**Error:** The condition `count = 0` is an assignment, not a comparison, causing a compilation error.

**Correction:** Change `count = 0` to `count > 0`.

```
public class IncorrectWhileCondition {
    public static void main(String[] args) {
        int count = 5;
        while (count > 0) { // Corrected: count > 0
```

```
            System.out.println(count);
            count--;
        }
    }
}
```

## Snippet 03

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

**Error:** The loop executes only once because `num > 0` becomes false after the first iteration.

**Correction:** Change the condition to `num <= 5` and initialize `num` to `1`.

```
public class DoWhileIncorrectCondition {
    public static void main(String[] args) {
        int num = 1;
        do {
            System.out.println(num);
            num++;
        } while (num <= 5); // Corrected: num <= 5
    }
}
```

## Snippet 04

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

**Error:** The loop prints numbers `1` to `10`, but if the expected output is `1` to `9`, there's an off-by-one error.

**Correction:** Change `i <= 10` to `i < 10`.

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

---

## Snippet 05

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

**Error:** The loop doesn't print numbers in the expected ascending order.

**Correction:** Initialize `i = 0`, condition `i <= 10`, and increment `i++`.

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

---

## Snippet 06

```
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:** "Done" prints only once outside the loop due to missing curly braces `{}`.

**Correction:** Enclose both statements within `{}`.

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

---

## Snippet 07

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

**Error:** `count` is declared but not initialized, causing a compilation error.

**Correction:** Initialize `count` to `0`.

```
public class UninitializedWhileLoop {
    public static void main(String[] args) {
        int count = 0; // Corrected: Initialized count
        while (count < 10) {
            System.out.println(count);
            count++;
        }
    }
}
```

---

## Snippet 08

```
public class OffByOneDoWhileLoop {
    public static void main(String[] args) {
```

```
        int num = 1;
        do {
            System.out.println(num);
            num--;
        } while (num > 0);
    }
}
```

**Error:** The loop prints unexpected numbers due to incorrect initialization and decrement operation.

**Correction:** Initialize `num = 1`, set condition `num <= 5`, and increment `num++`.

```
public class OffByOneDoWhileLoop {
    public static void main(String[] args) {
        int num = 1;
        do {
            System.out.println(num);
            num++;
        } while (num <= 5);
    }
}
```

---

## Snippet 09

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

**Error:** The loop skips numbers (prints `0, 2, 4` instead of `0, 1, 2, 3, 4`).

**Correction:** Increment `i` by `1` instead of `2`.

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

---

## Snippet 10

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

**Error:** The condition `num = 10` is an assignment, causing a compilation error.

**Correction:** Change `num = 10` to `num >= 0`.

```
public class IncorrectWhileLoopControl {
    public static void main(String[] args) {
        int num = 10;
        while (num >= 0) { // Corrected: num >= 0
            System.out.println(num);
            num--;
        }
    }
}
```

---

## 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:** The loop skips numbers (`0, 2, 4` instead of `0, 1, 2, 3, 4`).

**Correction:** Change `i += 2` to `i++`.

```
public class IncorrectLoopUpdate {
    public static void main(String[] args) {
```

```
        int i = 0;
        while (i < 5) {
            System.out.println(i);
            i++; // Corrected: 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:** x is declared inside the loop, making it inaccessible outside.

**Correction:** Declare x outside the loop.

```
public class LoopVariableScope {
    public static void main(String[] args) {
        int x = 0; // Corrected: Declared x outside the loop
        for (int i = 0; i < 5; i++) {
            x = i * 2;
        }
        System.out.println(x);
    }
}
```

---