

Lab Assignment 3

Loop Errors

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?
- **Output** - Infinite Loop.
- Because of decrement, i value goes in negative.

Corrected Code-

```
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?**
- **Getting compile time error-** rror: incompatible types: int cannot be converted to boolean
- while (count = 0)

Correced Code-

```

public class CorrectWhileCondition {

    public static void main(String[] args) {

        int count = 5;

        while (count > 0) {

            System.out.println(count);

            count--;

        }

    }

}

```

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 `do while` loop?
- **Getting Output-** Infinite loop

If num starts with 0 and increments to 1, the loop will continue indefinitely because num will always be greater than 0 after the first iteration.

Corrected Code-

```

public class DoWhileLimitedExecution {

    public static void main(String[] args) {

        int num = 0;

        do {

            System.out.println(num);

            num++;

        } while (num < 5); // Loop will stop when num is no longer less than 5

    }

}

```

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?**
- If we want the expected output the condition should be $i < 10$, we are giving $i \leq 10$ that is why it is printing 1 to 10. As i value is $10 = 10$ on 10th iteration and it becomes true so it prints 10 also.

Corrected Code-

```

public class OffByOneErrorForLoop {

    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?**
- In for loop , we need to decrement i values to get the expected order.
- for (int i = 10; i >= 0; i--)

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?**

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

- This loop will iterate from i = 0 to i = 4, printing the value of i each time.
- **System.out.println("Done");** ----> This line is outside the loop because there are no curly braces {} enclosing it. Therefore, it executes only once, after the loop completes.
- This line is outside the loop because there are no curly braces {}. It executes only once, after the loop completes.
- In Java, when a for loop (or any loop) is followed by a single statement without curly braces, only that single statement is considered part of the loop body.

Corrected code -

```
public class CorrectForLoopBody {  
  
    public static void main(String[] args) {  
  
        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?**
 - variable count might not have been initialized ---> while (count < 10)
 - initialize variable ----> int count = 4;
-

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?**
- **Output - 1**

num is initialized as and print 1 then it decrement to 0, the condition will become false.

Corrected Code-

```

public class OffByOneDoWhileLoop {

    public static void main(String[] args) {

        int num = 1;

        do {

            System.out.println(num);

            num++;

        } while (num <= 5);

    }

}

```

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?

- Because of this is `i += 2`, it is printing 0, 2, 4 ---> increment by 2. means `i = i + 2`.
 - correct ---> `for (int i = 0; i < 5; i ++)`
-

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?

- **Error:** incompatible types: int cannot be converted to boolean
 - In Java, we have to put a condition in while loop so that it can check where it is true or not. Because while condition accepts boolean value ----> `while (num > 10)`
-

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?

- Because of this is `i += 2`, it is printing 0, 2, 4 ---> increment by 2. means `i = i + 2`.

- correct ---> 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.

- The variable 'x' cause a compilation error because its scope is only in for loop. if we want to access x then we have to declare and initialize it outside the loop.
- int x = 0;

Guess the Output

Snippet 1:

```
public class NestedLoopOutput {
    public static void main(String[] args) {
        for (int i = 1; i <= 3; i++) {
            for (int j = 1; j <= 2; j++) {
                System.out.print(i + " " + j + " ");
            }
            System.out.println();
        }
    }
}
```

// i = 1, 2, 3
1st iteration // j = 1, 2, (3 => condition false),
2nd iteration // j = 1, 2, (3 => condition false),
3rd iteration // j = 1, 2, (3 => condition false)

1st iteration // 1 1 1 2
2nd iteration // 2 1 2 2
3rd iteration // 3 1 3 2

1st iteration //excutes after condition false (3<=2)
2nd iteration //excutes after condition false (3<=2)

Output-

```
1 1 1 2
2 1 2 2
3 1 3 2
```

Snippet 2:

```
public class DecrementingLoop {
    public static void main(String[] args) {
        int total = 0;

        for (int i = 5; i > 0; i--) {
            total = total + i;

            if (i == 3)
                continue;

            total = total - 1;
        }
        System.out.println(total);
    }
}
```

// i = 5, 4, 3, 2, 1, 0 then (i > 0 => condition False)
// total = 0+5 => 5, 4+4 => 8, 7+3 => 10,
10+2 => 12, 11+1 = 12

// True in 3rd iteration
// skips the remaining statements in this loop
iteration, so (total = total - 1) is not executed.

// total = 5-1 =>4, 8-1 =>7, not executed in 3rd
iteration, 12-1 =>11, 12-1 =>11

// 11

Output-

```
11
```

Snippet 3:

```
public class WhileLoopBreak {
    public static void main(String[] args) {
        int count = 0;
        while (count < 5) {

            System.out.print(count + " ");
            count++;
            if (count == 3)
                break;
        }
        System.out.println(count);
    }
}
```

// count = 0
// (0<5) => True
// (1<5) => True
// (2<5) => True

// 0 1 2
// 0++ =>1, 2, 3
// (1==3) => False, (2==2) => false, (3=3) => True
// loop is exited

//3

Output-

0 1 2 3

Snippet 4:

```
public class DoWhileLoop {
    public static void main(String[] args) {
        int i = 1;                                // i = 1
        do {
            System.out.print(i + " ");             // 1 2 3 4
            i++;                                    // 2, 3, 4, 5
        } while (i < 5);                           // (2<5) => True, (2<5) => True, (3<5) => True,
                                                    // (4<5) => True, (5<5) => False
        System.out.println(i);                     // 5
    }
}
```

Output-

1 2 3 4 5

Snippet 5:

```
public class ConditionalLoopOutput {
    public static void main(String[] args) {
        int num = 1;                               // num = 1
        for (int i = 1; i <= 4; i++) {              // i = 1, 2, 3, 4, 5
            if (i % 2 == 0) {                       // (1%2 = 0) False, True, False, True
                num += i;                           // num = num + i = 0 + 2 => 2, 3
            } else {
                num -= i;                           // num = num - i => 0, 2 - 3 => -1
            }
        }
        System.out.println(num);                   // 3
    }
}
```

Output-

3

Snippet 6:

```
public class IncrementDecrement {
    public static void main(String[] args) {
        int x = 5;                                // x = 5
        int y = ++x - x-- + --x + x++;            // y = 6 - 6 + 4 + 4
        System.out.println(y);                    // 8
    }
}
```

```
}
```

Output-
8

Snippet 7:

```
public class NestedIncrement {  
    public static void main(String[] args) {  
        int a = 10;                                // a = 10  
        int b = 5;                                // b = 5  
        int result = ++a * b-- --a + b++;          // result = 11 * 5 - 10 + 4  
        System.out.println(result);                // 49  
    }  
}
```

Output-
49

Snippet 8:

```
public class LoopIncrement {  
    public static void main(String[] args) {  
        int count = 0;                            // count = 0  
        for (int i = 0; i < 4; i++) {              // i = 0 true, (3 < 4 true), (6 < 4 false)  
            count += i++ - ++i;                    // for i = 0  
                                                    // count = count + i++ - ++i  
                                                    // 0 + 0 - 2 = -2 and here i becomes i = 2  
            1st iteration  
                                                    // for i = 3  
                                                    // -2 + 3 - 5 = -4 and here i becomes i = 5  
            2nd iteration  
        }  
        System.out.println(count);                // -4  
    }  
}
```

Output-
-4
