CDAC MUMBAI
Lab Assignment

SECTION 1: Error-Driven Learning Assignment: Loop Errors

Instructions:

Analyze each code snippet for errors or unexpected behavior. For each snippet, determine:

1. Why does the error or unexpected behavior occur?
2. How can the code be corrected to achieve the intended behavior?

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?
```

bcoz of decrement

we can use increment instead of decrement
or we can change condition also

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?
```

= is equivalent to assign the value and 0 is false
so it incompatible type that int to convert int boolean

Snippet 3:

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

loop we execute infinite times

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

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?
```

change condition to i<10 instead of i<=10

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?
```

instead of incrementing we can use decrement operator so output will be 10 to 0 in descending order

---

Snippet 6:

we are not defined the scope of for loop using {}
so it is on considering one statement is in for loop and
sop of done is only executing after working of for loop

```
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?
```

Snippet 7:

```
public class UninitializedWhileLoop {
    public static void main(String[] args) {
        int count;
```

int count =0;

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

initialization is not done

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

Snippet 8:

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

num++
and num<6 or num<=5

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

Snippet 9:

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

0
1
2
will be the output because we are incrementing the 'i' with 2

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

Snippet 10:

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

it has compilation error because in while loop condition we are assigning the number we can't assign the number in the condition

// 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?
```

0
1
2
will be the output because we are incrementing the 'i' with 2

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
```

x is declared in for loop so we cannot access outside the loop scope

SECTION 2: Guess the Output

Instructions:

1. **Perform a Dry Run:** Carefully trace the execution of each code snippet manually to determine the output.
2. **Write Down Your Observations:** Document each step of your dry run, including the values of variables at each stage of execution.
3. **Guess the Output:** Based on your dry run, provide the expected output of the code.
4. **Submit Your Assignment:** Provide your dry run steps along with the guessed output for each code snippet.

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
1 1 1 2
i=2
2 1 2 2
i=3
3 1 3 2

```
    }
  }
}
// Guess the output of this nested loop.
```

i = 5
total= 0+5=5 i!= 3
total= 5-1 =4

i=4
total=4+4=8 i!=3
total = 8-1=7

output =11

i=3
total=7+3=10 i==3
no subtraction

i=2
total= 10+2=12 i!=3
total=12-1=11

i=1
total=11+1=12 i!=3
total=12-1=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);
  }
}
// Guess the output of this while loop.
```

0
1 inside loop
2

count==3 loop will break

3 outside loop

Snippet 4:

```
public class DoWhileLoop {
  public static void main(String[] args) {
    int i = 1;
    do {
      System.out.print(i + " ");
      i++;
    } while (i < 5);
    System.out.println(i);
  }
}
// Guess the output of this do-while loop.
```

1
2 inside loop
3
4

5 outside loop

Snippet 5:

```
public class ConditionalLoopOutput {
   public static void main(String[] args) {
      int num = 1;
      for (int i = 1; i <= 4; i++) {
         if (i % 2 == 0) {
            num += i;
         } else {
            num -= i;
         }
      }
      System.out.println(num);
   }
}
// Guess the output of this loop.
```

i=1
1%2!=0
1-1=0

i=2
2%2==0
0+2=2

i=3
3%2!=0
2-3=-1

i=4
4%2==0
-1+4=3

output=3

Snippet 6:

```
public class IncrementDecrement {
   public static void main(String[] args) {
      int x = 5;
      int y = ++x - x-- + --x + x++;
      System.out.println(y);
   }
}
// Guess the output of this code snippet.
```

y=6-6+4+4
y=8

Snippet 7:

```
public class NestedIncrement {
   public static void main(String[] args) {
      int a = 10;
      int b = 5;
      int result = ++a * b-- - --a + b++;
      System.out.println(result);
   }
}
// Guess the output of this code snippet.
```

result= 11*5-10+4
=55-10+4
= 45+4
=49

Snippet 8:

```
public class LoopIncrement {
   public static void main(String[] args) {
      int count = 0;
      for (int i = 0; i < 4; i++) {
         count += i++ - ++i;
      }
      System.out.println(count);
   }
```

i=0
count=0+0-2=-2

i=2
count=-2+1-3=-4

output=-4