Assignment 3

Snippet 1:

```java
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?

Ans: The Loop goes on Infinitely because decrement is used in the code which will never be false for the loop to stop. So when the value for i pass through the loop it gets decremented for e.g here i = 0 so it will start going negative -1 which is less than the condition i.e i < 10 which will never occur in this case. So to solve this issue we will use increment in place of decrement.

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

```
C:\Users\ASUS\Desktop\Aug24CDAC\day3cdacpre>java InfiniteForLoop
0
1
2
3
4
5
6
7
8
9
```

Snippet 2:

```java
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 value passed through the while loop that is count = 0 does not satisfy the need for while loop. The while loop need a Boolean like true or false not literally but the condition should give answer to the while loop in terms of true or false which will decide the execution of the loop or not.

```
IncorrectWhileCondition.java:4: error: incompatible types: int cannot be converted to boolean
 while (count = 0) {
             ^
1 error
```

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

```
C:\Users\ASUS\Desktop\Aug24CDAC\day3cdacpre>java IncorrectWhileCondition
5
4
3
2
1
```

Snippet 3:

```java
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: Infinite Loop

Snippet no.4

```java
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: In this snippet the loop starts from I = 1 and then the condition is checked which is I<=10 so its true which is why 1 is printed and then I is incremented so, when I=10 it will also be executed because the condition says greater than or equal to so if we want the output to be from 1to 9 we just have to update the condition to I < 10 which will make it false and execution will be stopped.

```java
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
 }
}
```

```
C:\Users\ASUS\Desktop\Aug24CDAC\day3cdacpre>java OffByOneErrorForLoop
1
2
3
4
5
6
7
8
9
```

Snippet no.5

```java
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 this I is initialize at 10 so the execution will start from 10 and go on infinitely as there is no terminating condition as any value will be greater than 0.

Snippet no.6

```java
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?

Ans: As the syntax of for loop is not followed correctly it only considers only the next line after the for condition to contain the whole 2 statements we must enclose it in '{ }' paranthesis.

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

```
C:\Users\ASUS\Desktop\Aug24CDAC\day3cdacpre>java MisplacedForLoopBody
0
Done
1
Done
2
Done
3
Done
4
Done
```

Snippets no.7

```java
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: In this the count variable is not initialized and hence the compilation error. We can initialize count = 0 or whatever for the code to work.

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

```
C:\Users\ASUS\Desktop\Aug24CDAC\day3cdacpre>java UninitializedWhileLoop
0
1
2
3
4
5
6
7
8
9
```

Snippet no.8

```java
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: In this the number is decremented in each iteration but as the I = 1 it will print 1 and later on it will become 0 which is not fullfiling the condition that is less than 0. So to make it write we can set num = 1 and num <= 5 and num++ for desired output that is 1to5.

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

```
C:\Users\ASUS\Desktop\Aug24CDAC\day3cdacpre>java OffByOneDoWhileLoop
1
2
3
4
5
```

Snippet no.9

```
1  public class InfiniteForLoopUpdate {
2   public static void main(String[] args) {
3   for (int i = 0; i < 5; i += 2) {
4   System.out.println(i);
5   }
6   }
7  }
8
```

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

Ans: in this I is skipping 2 values as per the increment condition when i=0 it satisfy the condition and 0 is printed and then it gets incremented by 2 hence next value is 2 and then 4 and then it stops as condition turns false.

```
1  public class InfiniteForLoopUpdate {
2   public static void main(String[] args) {
3   for (int i = 0; i < 5; i++) {
4   System.out.println(i);
5   }
6   }
7  }
8
```

```
C:\Users\ASUS\Desktop\Aug24CDAC\day3cdacpre>java InfiniteForLoopUpdate
0
1
2
3
4
```

Snippet no.10

```
1  public class IncorrectWhileLoopControl {
2   public static void main(String[] args) {
3   int num = 10;
4   while (num = 10) {
5   System.out.println(num);
6   num--;
7   }
8   }
9  }
10
```

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

Ans :it will lead to a compilation error as in while loop num is being assigned a number which is not allowed.

Snippet no.11

```java
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: in this the output will be 0 2 4 as per the increment statement .we can change the increment to i++ for desired result.

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

```
C:\Users\ASUS\Desktop\Aug24CDAC\day3cdacpre>java IncorrectLoopUpdate
0
1
2
3
4
```

Snippet no.12

```java
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

Ans: in this x is declared and initializes in the for loop so its scope is limited to inside foe loop only outside of that it does not exist.

Section 2 : Guess the output

1. 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();
        }
    }
}
```

→ dry run

i	j	output	output
1	1	1 1	↓
	2	~~1~~ 1 2	1 1 1 2
	3		2 1 2 2
2	1	2 1	3 1 3 2
	2	~~2 2~~ 2 2	
	3		
3	1	3 1	
	2	~~3 2~~ 3 2	
	3		
4			

2. Snippet 2

```
public class DecrementationLoop {
    public static void main(String[] args) {
        int total = 0;
        for (int i = 5; i > 0; i--) {
            total += i;
            if (i == 3) continue;
            total -= 1;
        }
        SOP(total);
    }
}
```

→ dry run :-

total	i	total		output
0	5	5-1	4	11
	4	4+4-1	7	
	3	7+3	10	
	2	10+2-1	11	
	1	11+1-1	11	
	0			

3. Snippet 3

```
....
    int count = 0;
    while(count < 5){
        SOP(count++ " ");
        count++;
        if (count == 3) break;
    } }
    SOP ln (count);
```

PAGE No.
DATE / /

→ dry run

count

0
1
2
3

output

0 1 2 3

4. Snippet 4

```
....
    int i=1;
    do {
      SOP(i+" ");
      i++;
    } while(i<5);
      SOPln(i);
  }
}
```

→ dry run

i

1
2
3
4
5

output

1 2 3 4 5

1 2 3 4 5

5. Snippet 5

```
int num=1
for(int i=1; i<=4; i++){
    if(i%2==0){
        num+=i;
    } else {
        num -=i;
    }
}
SOPln(num)
}
}
```

→ dry run

num	i	num		output
1	1	1-1	0	
	2	0+2	2	3
	3	2-3	-1	
	4	-1+4	[3]	
	5			

6. Snippet no. 6

```
int x=5;
int y = ++x - x-- + --x + x++;
SOPln(y);
}
}
```

→ dry run

x	6-6+4+4 = 8
6	
5	
4	
5	

7. Snippet 7

```
....
int a = 10;
int b = 5;
int result = ++a * b-- - --a + b++;
SOPln(result);
```

→ dry run

a	b	11*5-10+4	output
11	4	55-10+4	49
10	5	49	

8. Snippet 8

```
....
int count = 0;
for(int i=0; i<4; i++){
    count += i++ - ++i;
}
SOPln(count);
}
```

→ dry run

count	i	0+0-2	output
0	0	-2+2-4	-4
	1	-4	
	2		
	4		