SECTION 1: Error-Driven Learning Assignment: Loop Errors

1.

Code:

```
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?

- Because i is decrementing infinitely.

How should the loop control variable be adjusted?

- We have to increment i that is i++.

Corrected Code:

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

2.

Code:

```
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?

- because count=0 means we are assigning 0 to the count. But here we have to compare values.

What is the issue with the condition in the `while` loop?

- here we have to compare values for that we can use comparison or relational operators. (==, >, < etc.)

Corrected Code:

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

3.

Code:

```
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?

- Loop is executing infinetly. It is because value of num is incrementing and that’s why it is always greater than 0.

What is wrong with the loop condition in the ` do while` loop?

- At one point condition should be false otherwise we will get infinite output. Here we can decrement n so the will loop will execute at once.

Corrected Code:

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



   }
}
```

4.

Code:

```
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?

- We want numbers from 1 to 9. but we have taken condition as i<=10 Which means loop will iterate 10 times including 10 because value of i is equal to 10 if we don't want consider 10 then we have to use i<10 which means 10 will not consider here because 10<10 will not satisfy and loop will execute for 1 to 9.

How should the loop be adjusted to meet the expected output?

- Here we have to remove = sign from condition i<=10 and make it i <10.

Corrected Code:

```
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
  }
```

5.

Code:

```
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?

- Because value of i is 10 and it is incrementing and it will increment infinitely because condition is i>=0 which means i will be always greater than 0.

What is the problem with the initialization and update statements in the `for` loop?

- At one point condition should be false otherwise we will get infinite output. Here we can decrement i so the will loop will execute from 10 to 0.

Corrected Code:

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

6.
Code:

```
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?

- After using for() loop we have to use {}. Then we write codes in that for loop() { .....code......). in this code, {} are not used and thats why for is considering the first line after loop as a part of for loop and other lines after that are outside the for loop like this:

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

  That’s why “Done” prints only once outside the loop.

How should the loop body be enclosed to include all statements within the loop?

- We have to use {} after for() and include all the statements that we want inside the loop write that inside that {}.

Corrected Code:

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

7.

Code:

```
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?

- Because count is not initialized

What needs to be done to initialize the loop variable properly?

- Initialize count by taking in any values like 0,1 etc.

Corrected Code:

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

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

8.
Code:

```
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?

- Because value of num is 1 and then it gets decremented to 0 which does not satisfy the condition num>0 .

What adjustments are needed to print the numbers from 1 to 5?

- Instead of decrementing, we have to increment num. Then we have to give (num<6) condition for while. So the value of num will increment from 1 to 5.

Corrected Code:

```
public class OffByOneDoWhileLoop {
  public static void main(String[] args) {
    int num = 1;
    do {
            System.out.println(num);
            num++;
        } while (num < 6);
    }
}
```

9.
Code:

```
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?

- This code is running well.

How should the loop update expression be corrected?

- If we want output as 0 1 2 3 4 (incrementing by 1) instead of 0 2 4 (incrementing by 2) then we have to change i+=2 to i+=1 or i++.

Corrected Code:

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

10.
Code:

```
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?

- Because in while the condition should be true or false. Here num = 10 is assigning value 10 to num. Instead of assigning we have to compare the input with this which we can do by using num == 10. If the input is similar to this num it means the condition is true and then it will iterate through the loop.

Corrected Code:

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

11.

Code:

```
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?

- 0 1 2 3 4
- This code is right if we want output of 0 2 4.

How should the loop variable be updated to achieve the desired result?

- I +=1

Corrected Code:

```
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
    }
  }
}
```

12.

Code:

```
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?

- We have to declare and initialize the x outside the or loop.

Corrected Code:

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

SECTION 2: Guess the Output

1.

Code:

```
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();
```

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

Dry run:

i	j	output
1	1 2	1 1 1 2
2	1 2	2 1 2 2
3	1 2	3 1 3 2

Guess:

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

2.

Code:

```
public class DecrementingLoop {
  public static void main(String[] args) {
    int total = 0;
    for (int i = 5; i > 0; i--) {
      total += i;
      if (i == 3) continue;
      total -= 1;
    }
    System.out.println(total);
  }
}
// Guess the output of this loop.
```

Dry run:

i	total
5	0 + 5 = 5
4	5 + 4 = 9
3	9 + 3 = 12 12 – 1 = 11
2	11 + 2 =13
1	13 + 1 = 14

Guess:
14

3.
Code:

```
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.
```

Dry run:

i	
0	0
1	1
2	2 3
3	
4	

Guess:

0 1 2 3

4.

Code:

```
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.
```

Dry run:

i	
1	1
2	2
3	3
4	4 5

Guess:

1 2 3 4 5

5.

Code:

```
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.
```

Dry run:

i	
1	num = 1 – 1 = 0
2	num = 0 + 2 = 2
3	num = 2 – 3 = -1
4	num = -1 + 4 = 3

Guess:

3

6.

Code:

```
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.
```

Dry run:

```
++x = 6
x-- = 6 => 5
--x = 4
x++ = 4 => 5

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

Guess:
8

7.
Code:

```
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.
```

Dry run:
++a = 11
b-- = 5 => 4
--a = 10
b++ = 4 => 5

11 * 5 – 10 + 4
= 49

Guess:
49

8.
Code:

```
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);
   }
}
// Guess the output of this code snippet.
```

Dry run:

i	Count = 0
0	0 + 0 - 2 = -2
1	-2 + 1 - 3 = -4
2	-4 + 2 - 4 = -6
3	-6 + 3 - 5 = -8

Guess:

-8

SECTION 3: Lamborghini Exercise:

1.Write a program to calculate the sum of the first 50 natural numbers.

```
class NaturalNum{
public static void main (String args[]) {
int sum =0;
for(int i=1; i<=50; i++) {
sum += i;
}
System.out.println("Sum of first 50 natural " + sum);
}
}
```

2. Write a program to compute the factorial of the number 10.

```
class Factorial {
public static void main (String args[]) {
int fact = 1;
for(int i=10; i>0; i--) {
fact = fact * i;
}
System.out.println("Factorial of 10 is " + fact);
}
}
```

3. Write a program to print all multiples of 7 between 1 and 100.

```
class MultipleOf7{
public static void main(String args[]){

for(int i=1 ; i<=100 ; i++){
if(i%7==0){
System.out.println(i + " is multiple of 7.");
}
}
}
}
```

4. Write a program to reverse the digits of the number 1234. The output should be 4321.

```
import java.lang.Math;
class Reverse{
public static void main(String args[]){
int num = 1234;
int y = 0;
for(int i=3 ; i>=0 ; i--){
int x = num%10;
y = y + x*((int)Math.pow(10,i));
num = num/10;

}
System.out.println("Reverse of 1234 is " + y);
}
}
```

5. Write a program to print the Fibonacci sequence up to the number 21.

```
class Fibonacci {
  public static void main(String[] args) {
        int n = 21, n1 = 0, n2 = 1;
        System.out.println("Fibonacci sequence up to " + n + ":");

        while (n1 <= n) {
            System.out.print(n1 + " ");
            int n3 = n1 + n2;
            n1 = n2;
            n2 = n3;
        }
    }
}
```

6. Write a program to find and print the first 5 prime numbers.

```
class Prime {
    public static void main(String[] args) {
        int count = 0;
        int num = 2;
        while (count < 5) {
            boolean Prime = true;
            for (int i = 2; i <= Math.sqrt(num); i++) {
                if (num % i == 0) {
                    Prime = false;
                    break;
                }
            }
            if (Prime) {
                System.out.println(num);
                count++;
            }
            num++;
        }
    }
}
```

7. Write a program to calculate the sum of the digits of the number 9876. The output should be 30 (9 + 8 + 7 + 6).

```
import java.lang.Math;
```

```
class Sum{
public static void main(String args[]){
int num = 9876;
int y = 0;
for(int i=3 ; i>=0 ; i--){
int x = num%10;
y = y + x;
num = num/10;

}
System.out.println("Sum of digits of num 9876 is " + y);
}
}
```

8. Write a program to count down from 10 to 0, printing each number.

```
class Countdown{
public static void main(String args[]){
int count = 0;
for(int i=10 ; i>=0 ; i--){
System.out.println(i);
System.out.println("Count : " + count);
count++;
}
}
}
```

9. Write a program to find and print the largest digit in the number 4825.

```
class Largenum{
public static void main(String args[]){
int num = 4825;
int y = 0;

for(int i=3 ; i>=0 ; i--){
int x = num%10;
if (x>y){
y = x;
}
num = num/10;

}
System.out.println("Largest digit of 4825 is " + y);
}
}
```

10. Write a program to print all even numbers between 1 and 50.

```
class Even{
public static void main(String args[]){
for(int i=1 ; i<=50 ; i++){
if (i%2==0){
System.out.println(i + " is Even");
}
}
}
}
```

11. Write a Java program to demonstrate the use of both pre-increment and post-decrement operators in a single expression.

```
class Que11{
public static void main(String args[]){

int a = 10;
int b = 20;

int sum1 = a + b;
++a;
++b;
int sum2 = a + b;
a--;
b--;
int sum3 = a + b;

System.out.println("Sum of two numbers : " + sum1);
System.out.println("Sum of two numbers after pre increment of a and b : " + sum2);
System.out.println("Sum of two numbers after post decrement of a and b : "
+ sum3);
}
}
```

12. Write a program to draw the following pattern:

```
*****
*****
*****
*****
*****
```

```
class Que12{
public static void main(String args[]){

for(int i=0 ; i<5 ; i++){
for(int j=0 ; j<5 ; j++){
System.out.print("*");
}
System.out.println(" ");
}
}
}
```

13. Write a program to print the following pattern:

```
1
2*2
3*3*3
4*4*4*4
5*5*5*5*5
5*5*5*5*5
4*4*4*4
3*3*3
2*2
1
```

```
class Que13{
```

```
public static void main(String args[]){
int count = 0;
for(int i=1 ; i<=5 ; i++){
for(int j=0 ; j<i ; j++){
if(j>0){
System.out.print("*");
}
System.out.print(count+i);
}
System.out.println(" ");
}
int count1 = 0;
for(int i=5 ; i>=1 ; i--){
for(int j=0 ; j<i ; j++){
if(j>0){
System.out.print("*");
}
System.out.print(count1+i);
}
System.out.println(" ");
}
}
}
```

14. Write a program to print the following pattern:

```
*
**
***
*****
*******
*********
```

```
class Que14{
public static void main(String args[]){



for(int i=1 ; i<=2 ; i++){
```

```
for(int j=0 ; j<i ; j++){
System.out.print("*");
}
System.out.println(" ");
}


for(int i=3 ; i<=9 ; i++){
if(i%2==0){
}
else{
for(int j=0 ; j<i ; j++){
System.out.print("*");
}
System.out.println(" ");
}
}




}
}
```

15. Write a program to print the following pattern:

```
    *
   **
  ***
 ****
*****
```

```
class Que15 {
   public static void main(String[] args) {
      for (int i = 1 ; i<=5 ; i++) {
         for (int j = 1 ; j<=5-i ; j++){
System.out.print(" ");
         }
         for (int k = 0 ; k<=i*1-1 ; k++){
```

```
System.out.print(" *");
            }
            System.out.println();
        }
    }
}
```

16. Write a program to print the following pattern:

```
    *
   ***
  *****
 *******
*********
```

```
class Que16 {
    public static void main(String[] args) {
        for (int i = 1 ; i<=5 ; i++) {
            for (int j = 1 ; j<=5-i ; j++){
System.out.print(" ");
            }
            for (int k = 1 ; k<=i*2-1 ; k++){
System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

17. Write a program to print the following pattern:

```
*****
 ****
  ***
   **
    *
```

```
class Que17 {
    public static void main(String[] args) {
      int n = 5;

      for (int i = n ; i>=1 ; i--) {
         for (int j = 1 ; j<=n ; j++)
if(j<=n-i){
System.out.print(" ");
}
else{
System.out.print(" *");
         }

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

18. Write a program to print the following pattern:

```
   *
  ***
 *****
*******
 *****
  ***
   *
```

```
class Que18 {
   public static void main(String[] args) {
      for (int i = 1 ; i<5 ; i++) {
         for (int j = 1 ; j<=5-i ; j++){
System.out.print(" ");
         }
         for (int k = 1 ; k<=i*2-1 ; k++){
System.out.print("*");
         }

         System.out.println();
      }
for (int i = 4 ; i>0 ; i--) {
         for (int j = 1 ; j<=6-i ; j++){
System.out.print(" ");
         }
         for (int k = 3 ; k<=i*2-1 ; k++){
System.out.print("*");
         }

         System.out.println();
      }



   }
}
```

19. Write a program to print the following pattern:
1
1*2
1*2*3
1*2*3*4
1*2*3*4*5

```
class Que19{
public static void main(String args[]){
int count = 0;
for(int i=2 ; i<=6 ; i++){
for(int j=1 ; j<i ; j++){
if(j>1){
System.out.print("*");
}
System.out.print(count+j);
}
System.out.println(" ");
}
}
}
```

20. Write a program to print the following pattern:

```
5
5*4
5*4*3
5*4*3*2
5*4*3*2*1
```

```
class Que20{
public static void main(String args[]){
int count = 6;
for(int i=2 ; i<=6 ; i++){
for(int j=1 ; j<i ; j++){
if(j>1){
System.out.print("*");
}
System.out.print(count-j);
}
System.out.println(" ");
}
}
}
```

21. Write a program to print the following pattern:

```
1
1*3
1*3*5
1*3*5*7
1*3*5*7*9
```

```
class Que21{
public static void main(String args[]){
int count =1;
for(int i=1 ; i<=9 ; i++){
if(i%2==0){
}
else{
```

```
for(int j=0 ; j<i ; j++){
int x = count +j;
if(x%2!=0){
if(j>1){
System.out.print("*");
}
System.out.print(x);
}
}
System.out.println(" ");
}
}
}
}
```

22. Write a program to print the following pattern:

```
*********
 *******
  *****
   ***
    *
   ***
  *****
 *******
*********
```

```
class Que22 {
   public static void main(String[] args) {

for (int i = 5 ; i>0 ; i--) {
        for (int j = 1 ; j<=5-i ; j++){
System.out.print(" ");
        }
        for (int k = 1 ; k<=i*2-1 ; k++){
System.out.print("*");
        }
        System.out.println();
```

```
      }

for (int i = 2 ; i<6 ; i++) {
        for (int j = 1 ; j<=5-i ; j++){
System.out.print(" ");
        }
        for (int k = 1 ; k<=i*2-1 ; k++){
System.out.print("*");
        }
        System.out.println();
      }

   }
}
```

23. Write a program to print the following pattern:

```
11111
22222
33333
44444
55555
```

```
class Que23{
public static void main(String args[]){

int count =1;
for(int i=1 ; i<=5 ; i++){
for(int j=1 ; j<=5 ; j++){
System.out.print(i);
}
System.out.println(" ");
}
}
}
```

24. Write a program to print the following pattern:

```
1
```

```
22
333
4444
55555
```

```
class Que24{
public static void main(String args[]){

int count =1;
for(int i=0 ; i<5 ; i++){
for(int j=0 ; j<=i ; j++){
System.out.print(count+i);
}
System.out.println(" ");
}
}
}
```

25. Write a program to print the following pattern:

```
1
12
123
1234
12345
```

```
class Que25{
public static void main(String args[]){

int count =1;
for(int i=0 ; i<5 ; i++){
for(int j=0 ; j<=i ; j++){
System.out.print(count+j);
}
System.out.println(" ");
}
}
}
```

26. Write a program to print the following pattern:

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

```
class Que26{
public static void main(String args[]){

int count =1;
for(int i=1 ; i<=5 ; i++){
for(int j=0 ; j<i ; j++){
System.out.print(count++ + " ");
}
System.out.println(" ");
}
}
}
```