Lab Assignment 3

SECTION 1: Error-Driven Learning Assignment: Loop Errors

Snippet 1:

Corrected code

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

Why does this loop run infinitely? Because there was i—so I never became greter than 10 hence it goes inti infinite loop.

How should the loop control variable be adjusted?

Snippet 2:

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

```
int count = 5;
while (count == 0) {
System.out.println(count);
count--;
}}}
```

Why does the loop not execute as expected? Because in while loop there should be two equal sign.

What is the issue with the condition in the `while` loop? In while loop there should be two equal sign.

Snippet 3:

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

Why does the loop only execute once? It is executing for infinite time.

What is wrong with the loop condition in the `dowhile` loop? Num in do loop should be decremented.

Snippet 4:

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

What is the issue with the loop boundaries? It execute one more no. thab expected.

How should the loop be adjusted to meet the expected output? In for loop condition will be i < 10 not i <= 10.

Snippet 5:

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

Why does this loop not print numbers in the expected order?

Because It should be in decrement not in increment.

What is the problem with the initialization and update statements in the `for` loop? It should be in decrement otherwise it will give us infinite loop.

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");
}
}
```

Why does "Done" print only once, outside the loop?

Because there is no curly brackets for the for loop .

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

Snippet 7:

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

Why does this code produce a compilation error? Need to initialize the count.

What needs to be done to initialize the loop variable properly? Need to use curly brackets

Snippet 8:

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

Why does this loop print unexpected numbers? Because num = 1 so it only print 1 as o/p

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

Snippet 9:

```
class InfiniteForLoopUpdate {
```

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

Why does the loop print unexpected results or run infinitely?

It is not running infinite .

How should the loop update expression be corrected? It is already giving correct o/p.

Snippet 10:

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

Why does the loop execute indefinitely? Because int cannot be converted to Boolean.

What is wrong with the loop condition? It should be like num>=10.

Snippet 11:

```
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
}
}
}
```

What will be the output of this loop? 0 2 4

How should the loop variable be updated to achieve the desired result?we can use i++ to get u/p as 1 2 3 4

Snippet 12:

```
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
}
}
```

Why does the variable 'x' cause a compilation error?

Because it is initialize in loop so it will be only valid for loop not outside the loop.

SECTION 2:

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();
}
}
}
```

```
o/p : 1112      dry run :  i    j
      2122                 1    1         1 1 1 2
      3132                      2
                           2    1         2 1 2 2
                                2
                           3    1         3 1 3 2
                                2
```

Snippet 2:

```
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);}}
```

```
O/p : 11        dry run:-  i    total          when we get i =3
                           5    5              then that value
                           4    9(5+4)         directly goes into
```

3 11(9+3-1) if loop so it won’t be able to turn back to for loop. Hence Total = 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);
}}
```

o/p: 0 1 2 3

dry run:-

count	o/p
0	0
1	1
2	2
3	3

Initial value of count is zero, in while loop it get incremented by 1 so whenever we get count as 3, our program will get terminated because there is break and it will only print 0 1 2 3 as output. because of print statement it will print in 1 single line.

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);
}}
```

o/p: 1 2 3 4 5

dry run:-

i	o/p
1	1
2	1 2
3	1 2 3
4	1 2 3 4
5	1 2 3 4

Initial value of i = 1 it get increment by 1 in a loop but there is one condition that i should be less than 5. Still we get no. 5 in output. Because when we get i as 5 our loop will break and there is one sop outside of a loop. That’s why we get 1 2 3 4 5 as output.

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);
}}
```

o/p:- 3

dry run:-

i	num
1	0
2	2
3	-1
4	3

In for loop i = 1 to 4 but if I is divisible by by 2 then num will get incremente by i th value otherwise it will be decrement by i th value. So at the end we get num = 3 (4 - 1). Hence the output

Snippet 6:

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

o/p:- 8 (6 – 6 + 4 + 4)

dry run:-

++x = 6   (it’s incremented by 1)

x-- = 6   (it will change in next x value now it is 6)

--x   = 4   (because in previous x—it become 5 and now it become 4)

X++ = 4 (it will change in next x value as 5 but now it is 4 only)

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);
}}
```

o/p:- 49 (11 * 5 – 10 + 4)

dry run:-

++a = 11 (It is incremented by 1)

b-- = 5 (now it is same as initial but in next b’s value it will be 4)

--a = 10 (it is decrement by 1 means 11-1 = 10)

b++ = 4 (in previous expression it became 4 so now it will be only 4 but in next b’s value it will be 5)

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);
}}
```

o/p:- -4

in last step I become 4 so it is less than 4 hence for loop get terminated.

dry run:-

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

2 -4 3 5 -2 + 3 – 5 = -4

SECTION 3: Lamborghini Exercise:

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

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

O/P :- 1275

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

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

```
O/P :-   3628800
```

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

```
class Main
{
	public static void main(String args[])
	{
		for(int i=1; i<=100; i++)
		{
			if(i % 7 == 0)
			{
				System.out.print(i + “ “);
			} }}}
```

```
o/p:-	7 14 21 28 35 42 49 56 63 70 77 84 91 98
```

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

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

```
    {
        int num = 1234;
        int reverceNum = 0;
        while(num != 0)
        {
            int digit = num % 10;
            reverceNum = reverceNum * 10 +
digit;
            num /= 10;
        }
        System.out.println("Reverce Number of
1234 is : " + reverceNum);
    }
}
```

```
o/p:-  Reverce Number of 1234 is : 4321
```

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

```
class Fibonachi
```

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

    int n = 21, firstTerm = 0, secondTerm = 1;
    System.out.println("Fibonacci Series till " + n + "
terms:");

    for (int i = 1; i <= n; ++i)
      {
     System.out.print(firstTerm + ", ");
     int nextTerm = firstTerm + secondTerm;
     firstTerm = secondTerm;
     secondTerm = nextTerm;
      }
  }
}
```

O/P:- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765

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

```
class Prime
{
	public static void main(String[] args)
	{
		for(int i=2; i<12; i++)
		{
			int count = 0;
			for(int j=2; j<i; j++)
			{
				if(i % j == 0)
				{
					count = 1;
				}
			}
			if(count == 0)
			{
```

```
                System.out.print(i + " ");
            }
        }

    }
}
```

O/P :- 2 3 5 7 11

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

```
class Main
{
    public static void main(String[] args)
    {
        int n = 9876,r,sum=0;
        while(n>0)
        {
            r = n % 10;
            sum += r;
            n /= 10;
        }
        System.out.println("Sum of number 9876 (9 + 8 + 7 + 6) is : " +sum);

    }
}
```

O/P:- Sum of number 9876 (9 + 8 + 7 + 6) is : 30

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

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

```
o/p:-
10
9
8
7
6
5
4
3
2
1
0
```

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

```
class Prime
{
	public static void main(String[] args)
	{
		int n = 4825;
		int max = 0, r;
		while(n>0)
		{
			r = n % 10;
			max = Math.max(r, max);
			n /= 10;
		}
		System.out.println("Largest digit of
number 4825 is : " + max);


	}
}
```

```
O/P:- Largest digit of number 4825 is : 8
```

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

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

```
o/p:- 2  4  6  8  10  12  14  16  18  20  22  24  26  28
30  32  34  36  38  40  42  44  46  48  50
```

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

```
class Prime
{
    public static void main(String[] args)
    {
        int a = 20;
        int b = 30;
        int c = a++  +  ++b  +  ++a  - a-- + --b;
        System.out.println(c);
    }
}
```

O/P:- 81

12. Write a program to draw the following pattern:

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

```
class Main
{
    public static void main(String args[])
    {
        for(int i=1; i<=5; i++)
        {
            for(int j=1;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 Main

{

    public static void main(String args[])

    {

        for(int i=1; i<=5; i++)

        {

            for(int j=1;j<i;j++)

            {

                    System.out.print(i + "*");

            }

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

14. Write a program to print the following pattern:

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

```
class Main
{
    public static void main(String args[])
    {
        for(int i=1; i<=10; i++)
        {
            if(i==5 || i==7 || i==9)
                {
                    continue;
                }
            for(int j=1;j<i;j++)
            {
                System.out.print("*");

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

15. Write a program to print the following pattern:

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

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

16. Write a program to print the following pattern:

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

```
class Main
{
        public static void main(String args[])
        {
                for(int i=0; i<=9; i++)
                {
                        if(i%2 != 0)
                        {
                        for(int j=9; j>i; j--)
                        {
                                System.out.print(" ");
                        }
                        for(int j=1;j<=i;j++)
                        {
                                        System.out.print(" *");
                        }
                        }
                        System.out.println();
                }
        }
}
```

17. Write a program to print the following pattern:

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

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

18. Write a program to print the following pattern:

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

```
class Main
{
        public static void main(String args[])
        {
                for(int i=0; i<=7; i++)
                {
                        if(i%2 != 0)
                        {
                        for(int j=7; j>i; j--)
                        {
                                System.out.print(" ");
                        }
                        for(int j=1;j<=i;j++)
```

```
    {
            System.out.print(" *");
    }
    }
    System.out.println();
}
for(int i=7; i>0; i--)
{
    if(i%2 != 0)
    {
        if(i==7)
        {
            continue;
        }
    for(int j=7; j>i; j--)
    {
        System.out.print(" ");
    }
    for(int j=1;j<=i;j++)
    {
            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 Main
{
    public static void main(String args[])
    {
        for(int i=1; i<=5; i++)
        {
            for(int j=1;j<i;j++)
            {
                    System.out.print(j + "*");
            }
```

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

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

22. Write a program to print the following pattern:

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

```
class Main
{
        public static void main(String args[])
        {
                for(int i=9; i>0; i--)
                {
                        if(i%2 != 0)
                        {
```

```
        for(int j=9; j>i; j--)
        {
                System.out.print(" ");
        }
        for(int j=1;j<=i;j++)
        {
                        System.out.print(" *");
        }
        }
        System.out.println();
}
for(int i=3; i<=9; i++)
{
        if(i%2 != 0)
        {
        for(int j=9; j>i; j--)
        {
                System.out.print(" ");
        }
        for(int j=1;j<=i;j++)
        {
                        System.out.print(" *");
```

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

23. Write a program to print the following pattern:

```
11111
22222
33333
44444
55555
```

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

24. Write a program to print the following pattern:

```
1
22
333
4444
55555
```

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

25. Write a program to print the following pattern:

1

12

123

1234

12345

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

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 Main
{
    public static void main(String args[])
    {
        int count = 0;
        for(int i=1; i<6; i++)
        {
            for(int j=0; j<i; j++)
            {
                System.out.print(++count);
            }
            System.out.println();
        }}}
```