SECTION 1:

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?
```

The i - - decrements the value from 0 to negative integers.to control the loop we can use i++.so it will print numbers from 0 to 9

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?
```

```
 error: incompatible types: int cannot be converted to boolean
        while (count = 0) {
                       ^
```

Instead of = which is assignment operator use == which is use to compare the values error gets solved but there is no output.if we change count==5 in sop the output is 5.or if we use count>0 count - - will decrement the value

Solved code:

```
class IncorrectWhileCondition {
    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?
```

Following loop results in infinite 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?
```

We get the output from 1 to 10 but to get actual output of 1 to 9. We remove the ‘=’ in the for loop

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?
```

Code runs in the loop.the i is incrementing indefinitely.

Correct code:

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

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?

Error: variable count might not have been initialized

```
        while (count < 10) {
              ^
```

Correct code:

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


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

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?

```
 error: class InfiniteForLoopUpdate is public, should be declared in a file named
InfiniteForLoopUpdate.java
public class InfiniteForLoopUpdate {
       ^
```

Code gives output of 0 2 4 and 1 3 are missing so we can replace i+=2 with i++ so we get output of 01234

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?

Correct code:

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

```
}
```

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

```
        while (num = 10) {
                 ^
```

Solution: num=10 is assignment operator so we should use n==10 to compare values

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?
```

Here the op is 0 2 4 because of i+=2 . we can solve this by either replacing i+=2 with i+=1 or by i++

Correct code:

```
class IncorrectLoopUpdate {
    public static void main(String args[]) {
        int i = 0;
        while (i < 5) {
            System.out.println(i);
```

```
            i += 1; // Error: This may cause unexpected results in output
        } }}
```

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
```

```
 error: cannot find symbol
        System.out.println(x); // Error: 'x' is not accessible here
                           ^
  symbol:   variable x
  location: class LoopVariableScope
```

Solution:this can be solve by moving the sop statement in the forloop

```
class LoopVariableScope1 {
    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
    }
}
```

SECTION 3:

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

```
class sumOfNaturalNo{
	public static void main(String args[]){
		int n = 50;
		int result = n*(n+1)/2;

		System.out.println("Sum of first 50 natural number is"+ result);
	}
}
```

Output:1275

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

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

		int i,fact=1;
		int number=10;

		for(i=1;i<=number;i++){

		fact=fact*i;

	}

	System.out.println("factorial of number 10 is" + fact);
}
}
```

o/p-3628800

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 = 7; i <= 100; i += 7) {
        System.out.print(i + " ");
      }
            }
        }
```

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

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

        {
            int number = 1234, reverse = 0;
            for( ;number != 0; number=number/10)
            {
                int remainder = number % 10;
                reverse = reverse * 10 + remainder;
                }
                System.out.println("The reverse of the given number is: " + reverse);
                }
                }
```

O/p- 4321

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, firstNum = 0, secondNum = 1;
   System.out.println("Fibonacci Series till " + n + " numbers:");

   for (int i = 1; i <= n; ++i) {
     System.out.print(firstNum + ", ");
     int nextNum = firstNum + secondNum;
     firstNum = secondNum;
     secondNum = nextNum;
   }
  }
}
```

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[]) {

  {
    int i, j, num, count=0;

    System.out.println("First 5 Prime Numbers are:");
    for(i=2; count<5; i++)
    {
      num = 0;
      for(j=2; j<i; j++)
      {
        if(i%j==0)
        {
          num++;
          break;
        }
      }
      if(num==0)
      {
        System.out.print(i+ " ");
        count++;
      }
    }
  }
}
```

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 SumOfDigits {
   public static void main(String args[]) {
      int number = 9876;
      int sum = 0;

      // Sum the digits in a single line using a loop
      for (; number > 0; number /= 10) {
```

```
            sum += number % 10;
        }

        System.out.println("The sum of the digits is: " + sum);
    }
}
```

O/p: 30

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

```
class Countdown {
    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 21 0

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

```
class LargestNumber {
    public static void main(String args[]) {
        int number = 4825;
        int largestDigit = 0;

        for (; number > 0; number /= 10) {

            int digit = number % 10;

            if (digit > largestDigit) {
                largestDigit = digit;
            }
        }

        System.out.println("The largest digit is: " + largestDigit);
    }
}
```

O/p: 8

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

```
class evenNumbers {
    public static void main(String args[]) {
        for (int i = 1; i <= 50; i++) {
            if (i % 2 == 0) {
                System.out.println(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 PreIncrementPostDecrement {
   public static void main(String args[]) {
      int a = 5;
      int b = 10;

      int result1 = ++a + b--;
                  int result2 = a++ + --b;

      System.out.println("Result of the expression: " + result1);
                  System.out.println("Result of the expression: " + result2);

      System.out.println("Value of a after expression: " + a);
      System.out.println("Value of b after expression: " + b);
   }
}
```

O/p: Result is: 16
Result is: 14
Value of a after expression: 7
Value of b after expression: 8

12. Write a program to draw the following pattern:

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

```
class StarPattern {
   public static void main(String args[]) {
      int rows = 5;
      int columns = 5;

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

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

13. Write a program to print the following pattern:

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

      for (int i = 1; i <= 2 * n - 1; i++) {
         int current = (i <= n) ? i : (2 * n - i);

         for (int j = 1; j <= current; j++) {
            System.out.print(current);
            if (j < current) {
               System.out.print("*");
            }
         }
         System.out.println();
      }
   }
```

O/p:

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

14.

```
class Pattern{
        public static void main(String args[]){
int n = 9;
                for(int i = 0; i < n; i++){
                for(int j = 0; j <= i; j++) {
                System.out.print("* ");
```

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

        }}
```

O/p:

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

15. question

```
public class Star {

    public static void printTriangle(int n) {
        // Loop for each row
        for (int i = 1; i <= n; i++) {
            // Print leading spaces
            for (int j = n - i; j > 0; j--) {
                System.out.print(" ");
            }

            // Print stars with spaces in between
            for (int j = 0; j < i; j++) {
                System.out.print("* ");
            }

            // Move to the next line
            System.out.println();
        }
    }

    public static void main(String[] args) {
        int n = 5; // Number of rows for the triangle
        printTriangle(n); // Call the method to print the triangle
    }
```

```
}
```

O/p:

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

16.

```
class Pattern {
    public static void main(String[] args) {
        int n = 5; //  rows
        for (int i = 1; i <= n; i++) {

            for (int j = i; j < n; j++) {
                System.out.print(" ");
            }


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

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

O/p:

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

17. class reversepattern{

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

   int number = 5;
   int i, j;

   for(i = number; i >= 1; i--)
   {

      for(j = i; j < number; j++)
      {
         System.out.print(" ");
      }

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

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

O/p:

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

18. class DiamondPattern {

```
   public static void main(String args[]) {
      int n = 4;

      // Upper half
      for (int i = 1; i <= n; i++) {
         for (int j = i; j < n; j++) {
            System.out.print(" ");
         }
         for (int j = 1; j <= (2 * i - 1); j++) {
```

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

    // Lower half
    for (int i = n - 1; i >= 1; i--) {
        for (int j = n; j > i; j--) {
            System.out.print(" ");
        }
        for (int j = 1; j <= (2 * i - 1); j++) {
            System.out.print("*");
        }
        System.out.println();
    }
  }
}
```

O/p:

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

19.
```
public class NumericPattern {
  public static void main(String[] args) {
    int n = 5; // Number of rows

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

O/p:

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

20.

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

        for (int i = 1; i <= n; i++) {
            int num = 1; //  first odd number
            for (int j = 1; j <= i; j++) {
                System.out.print(num);
                if (j < i) {
                    System.out.print("*");
                }
                num += 2; //  next odd number
            }
            System.out.println();
        }
    }
}
```

O/p:

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

23.

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

        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= n; j++) {
                System.out.print(i);
```

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

O/p:

```
11111
22222
33333
44444
55555
```

24.

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

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

O/p:

```
1
22
333
4444
55555
```

25.

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

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

O/p:

```
1
12
123
1234
12345
```