Java Notes

What is Java?

- ★ Java is a high-level, class-based, object-oriented programming language.
- ★ It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA)
- ★ Java was originally developed by James Gosling at Sun Microsystems.
- ★ It was released in May 1995.
- ★ Now, it is owned by Oracle Corporation.

James Gosling

Java Logo

History of Java:

- ★ Java programming language was initially called Oak after an oak tree that stood outside Gosling's office.
- ★ Later the project went by the name Green.
- ★ Finally renamed Java, from Java coffee, a type of coffee from Indonesia.
- ★ Sun Microsystems released the first public implementation as Java 1.0 in 1996.
- ★ Oracle Corporation's acquisition of Sun Microsystems happened 2009–10.
- ★ On April 2, 2010, James Gosling resigned from Oracle.Now, he is working in Amazon Web Services.

Version history of Java:

Version	Date
JDK Beta	1995
JDK 1.0	January 23, 1996
JDK 1.1	February 19, 1997
J2SE 1.2	December 8, 1998
J2SE 1.3	May 8, 2000
J2SE 1.4	February 6, 2002
J2SE 5.0	September 30, 2004
Java SE 6	December 11, 2006
Java SE 7	July 28, 2011
Java SE 8 (LTS)	March 18, 2014
Java SE 9	September 21, 2017
Java SE 10	March 20, 2018
Java SE 11 (LTS)	September 25, 2018
Java SE 12	March 19, 2019
Java SE 13	September 17, 2019
Java SE 14	March 17, 2020
Java SE 15	September 15, 2020
Java SE 16	March 16, 2021
Java SE 17 (LTS)	September 14, 2021
Java SE 18	March 22, 2022
Java SE 19	September 20, 2022
Java SE 20	March 21, 2023
Java SE 21 (LTS)	September 19, 2023

What is the LTS Version?

- ★ Long-term support (LTS) is a product lifecycle management policy.
- ★ Stable release of computer software is maintained for a longer period of time than the standard edition, which is typically called Short-term support (STS)

```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\chinn>java -version
openjdk version "21" 2023-09-19
OpenJDK Runtime Environment (build 21+35-2513)
OpenJDK 64-Bit Server VM (build 21+35-2513, mixed mode, sharing)
```

What is Open JDK?

- ★ OpenJDK (Open Java Development Kit) is a free and open-source implementation of the Java Platform, Standard Edition (Java SE).

Warm Up before we go into topics:

1- WAP to print “Hello World!” in java

```
public class Test {
   public static void main(String[] args) {
       System.out.println("Hello World!");
   }
}
```

Output:

```
Hello World!
```

2- WAP to add two numbers in Java

```
public class AddNumbers {
   public static void main(String[] args) {
       int num1 = 10;
       int num2 = 20;
       int result = num1 + num2;
       System.out.println("num1 + num2 = " + result);
   }
}
```

Output:

num1 + num2 = 30

3- WAP to find the area of rectangle in Java

```
public class AreaOfRectangle {
   public static void main(String[] args) {
       int length = 10;
       int breadth = 20;
       int area = length * breadth;
       System.out.println("Area of rectangle is :: " + area);
   }
}
```

Output:

Area of rectangle is :: 200

4- WAP to find sum of numbers in Java

```
public class SumOfNumbers {
   public static void main(String[] args) {
        int n = 10;
        int sum1 = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10;
        System.out.println("sum1 :: " + sum1);

        int sum2 = 0;
        for (int i = 1; i <= n; i++) {
            sum2 = sum2 + i;
        }
        System.out.println("sum2 :: " + sum2);

        int sum3 = 0;
        int count = 1;
        while (count <= n) {
            sum3 = sum3 + count;
            count++;
        }
        System.out.println("sum3 :: " + sum3);
   }
}
```

Output:

sum1 :: 55
sum2 :: 55
sum3 :: 55

Dry Run:

Iteration	Value of i	Logic(sum2+=i)	sum2
1	1	0+1	1
2	2	1+2	3
3	3	3+3	6
4	4	6+4	10
5	5	10+5	15
6	6	15+6	21
7	7	21+7	28
8	8	28+8	36
9	9	36+9	45
10	10	45+10	55

Matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \end{bmatrix}$$

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)

Array index starts from 0

5- WAP to print print above matrix #1

```
public class PrintMatrix {
   public static void main(String[] args) {
       int[][] matrix = new int[][]{
               {1, 2, 3},
               {4, 5, 6},
               {1, 2, 3}
       };
       for (int i = 0; i < 3; i++) {
           for (int j = 0; j < 3; j++) {
               System.out.print(matrix[i][j] + " ");
           }
           System.out.println();
       }
   }
```

Output:

1 2 3
4 5 6
1 2 3

6- WAP to print print above matrix #2

```
public class PrintMatrixIndices {
   public static void main(String[] args) {
       int[][] matrix = new int[3][3];

       //Print matrix
       for (int i = 0; i < 3; i++) {
           for (int j = 0; j < 3; j++) {
               System.out.print(matrix[i][j] + " ");
           }
           System.out.println();
       }
```

```
        //Print indices
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print("(" + i + "," + j + ")");
            }
            System.out.println();
        }
    }
}
```

Output:

```
0 0 0
0 0 0
0 0 0
(0,0)(0,1)(0,2)
(1,0)(1,1)(1,2)
(2,0)(2,1)(2,2)
```

7- WAP to print print above matrix #2

```
public class MatrixDiagonalSum {
    public static void main(String[] args) {
        int[][] matrix = new int[][]{
                {1, 2, 3},
                {4, 5, 6},
                {1, 2, 3}
        };
        int sum = 0;
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix.length; j++) {
                if (i == j) {
                    sum += matrix[i][j];
                }
            }
        }
        System.out.println("diagonal sum :: " + sum);
    }
}
```

Output:

```
diagonal sum :: 9
```

8- WAP to print a square pattern:

```
public class MatrixPattern1 {
    public static void main(String[] args) {
        int[][] matrix = new int[4][4];
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix.length; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

Output:

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

9- WAP to print a square pattern 2:

```
public class MatrixPattern2 {
    public static void main(String[] args) {
        int[][] matrix = new int[4][4];
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix.length; j++) {
                if (j <= i) {
                    System.out.print("* ");
                }
            }
            System.out.println();
        }
    }
}
```

Output:

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

9- WAP to print a square pattern 3:

```
public class MatrixPattern3 {
    public static void main(String[] args) {
        int[][] matrix = new int[4][4];
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix.length; j++) {
                if (i <= j) {
                    System.out.print("* ");
                }
            }
            System.out.println();
        }
    }
}
```

Output:

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

10- WAP to print a square pattern 4:

```
public class MatrixPattern4 {
    public static void main(String[] args) {
        int[][] matrix = new int[4][4];
        int i,j;
        int n = matrix.length;

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

Output:

```
    1
  2 1 2
 3 2 1 2 3
4 3 2 1 2 3 4
```

11- WAP to print a square pattern 4:

```
public class MatrixPattern5 {
   public static void main(String[] args) {
        int[][] matrix = new int[4][4];
        int i,j;
        int n = matrix.length;

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

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

Output:

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