INTRODUCTION TO JAVA

What is *Pseudo Code* ?

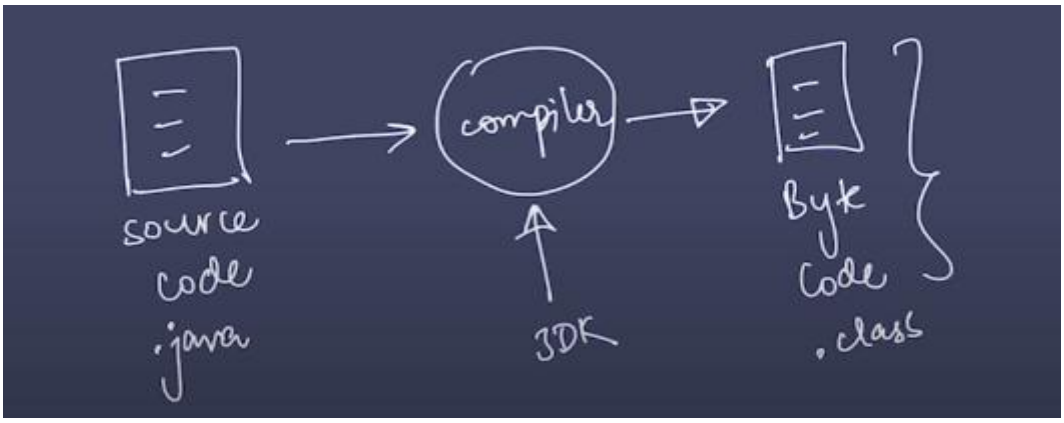
- Pseudocode is a description of an algorithm or program that uses natural language instead of a programming language.
- It's a blueprint for writing code, and it's often used as a rough draft.
- Basically converting programming language into English language that is understand by the user.

How my java Code is Running?

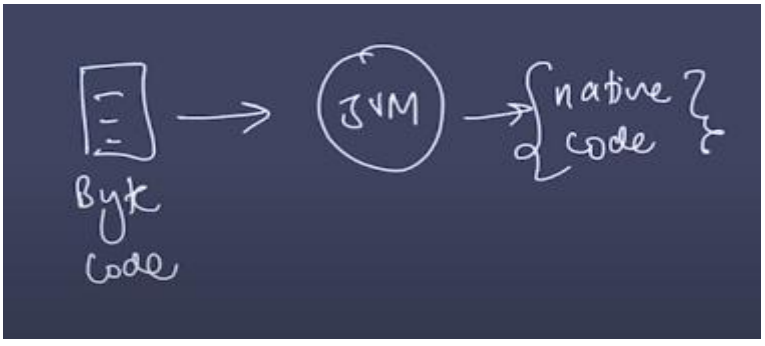
There are two steps from which my code is running

1. **Compilation**
2. **Execution**

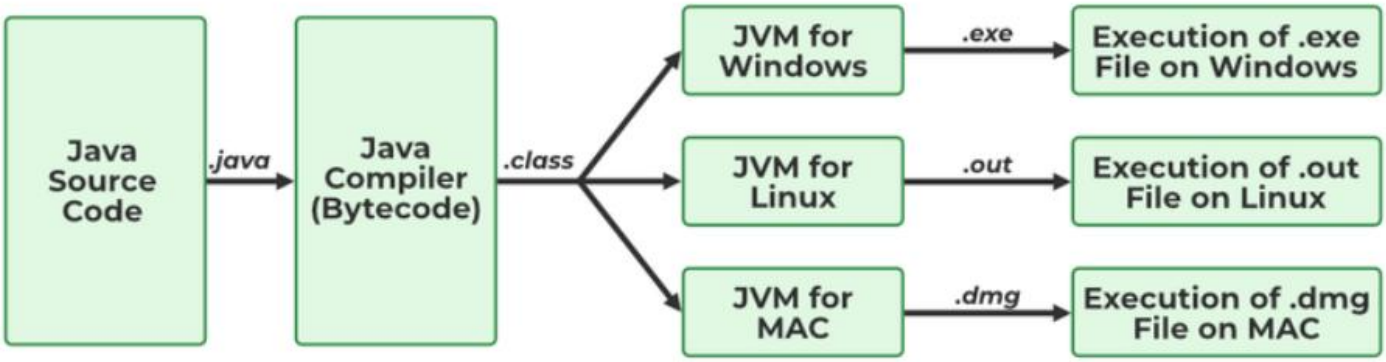
What happens in the compilation stage?



What happens in the Execution stage?



Together



What does JVM stands for what it is?

- JVM stands for Java Virtual Machine it is a Java interpreter.
- It is responsible for loading, verifying, and executing the bytecode created in Java.

What does JRE stand for and what it is?

- The Java Runtime Environment (JRE) is software that allows Java programs to run on a computer.
- It's a collection of tools that includes the Java Virtual Machine (JVM), Java class libraries, and the Java class loader.
- The JRE act as a translator between Java programs and the operating system, ensuring compatibility.

What does JDK stands for and what it does?

- JDK (Java Development Kit) stands for Java Development Kit.
- It is a software development kit used to develop Java applications.

Difference between JDK, JRE, and JVM

- JDK: Java Development Kit is a software development environment used for developing and Executing Java applications and applets.
- JRE: JRE stands for Java Runtime Environment and it provides an environment to run only the Java program onto the system. (not to deploy)

- JVM: JVM stands for Java Virtual Machine and is responsible for executing the Java program. (also called as interpreter)

BASIC FUNDAMENTALS IN JAVA

- **OUTPUT**
- **VARIABLES**
- **DATA TYPES**
- **INPUT**

Input/output

Syntax :- System.out.print(" ");

1. **print :- system.out.print("....");** the line continues
2. **println :- system.out.println("...");** the 2nd output goes to nextline
3. **"\n":-** goes to next line

VARIABLES

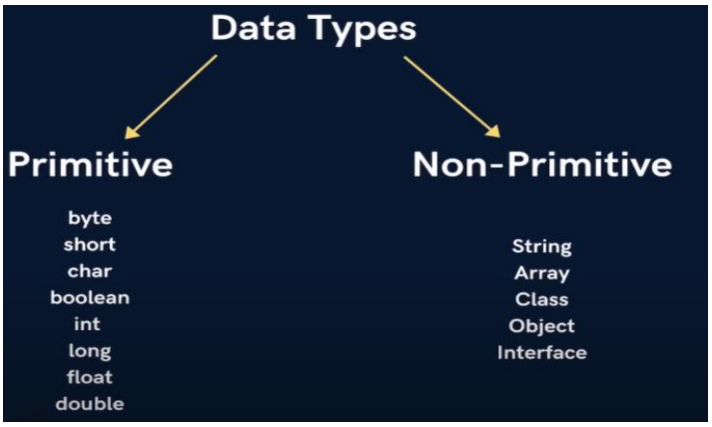
A variable is a named location in computer memory that stores data that can change.

How variables are stored in the computer memory?

1. Variables are stored in memory cells built from transistors and other components on an integrated circuit.
2. Variables are assigned a memory address when they are created.
3. The memory address is the location of the variable in the computer's memory.
4. The value of a variable is stored in the memory address.

DATATYPES

Since JAVA is a **Typed Language** we have to mention what is the type of data we have written to get him the proper space in the memory



- JAVA does not follow the BODMAS rule.
- In Java calculation happens from left to right.

INPUT

Eg code

package Input_03; → Declares that `Question.java` belongs to the `Input_03` package, which organizes classes and prevents naming conflicts; the file must be inside the `Input_03` folder for correct compilation.

import java.util.*; → Imports the `java.util` package, which includes the `Scanner` class used for taking user input.

public class Question { → Declares a public class `Question`, which must match the filename `Question.java`.

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

- This is the entry point of the Java program.
- `public` → Allows access from anywhere.
- `static` → Belongs to the class, doesn't need an object to be called.
- `void` → No return value.
- `String[] args` → Command-line arguments (not used here).

```
    Scanner sc = new Scanner(System.in);
```

- `Scanner sc` → Creates a `Scanner` object named `sc`.
- `new Scanner(System.in)` → Reads input from the keyboard.

```
    System.out.println("Enter two numbers : ");
    int a = sc.nextInt();
    int b = sc.nextInt();
```

- `sc.nextInt();` → Reads an integer input from the user and stores it in `a`.
- `sc.nextInt();` → Reads another integer input and stores it in `b`.

```
    int sum = a + b;
    System.out.println("Sum : " + sum);
    sc.close();
```

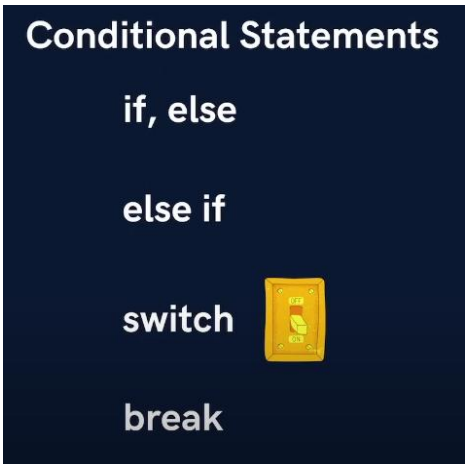
→ Closes the `Scanner` to free up system resources.

```
  }
}
```

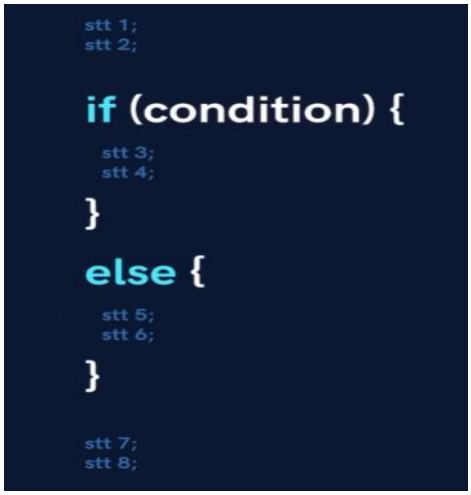
CONDITIONAL STATEMENTS

Conditional statements control the flow of a program based on certain conditions (true or false).

The main Conditional Statements are



1. If, Else



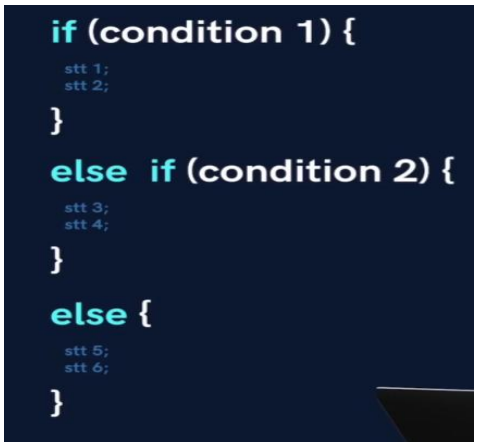
- Executes one block if the condition is `true`, otherwise executes another block.

```
if (age >= 18) {
    System.out.println("You can vote.");
} else {
    System.out.println("You cannot vote.");
}
```

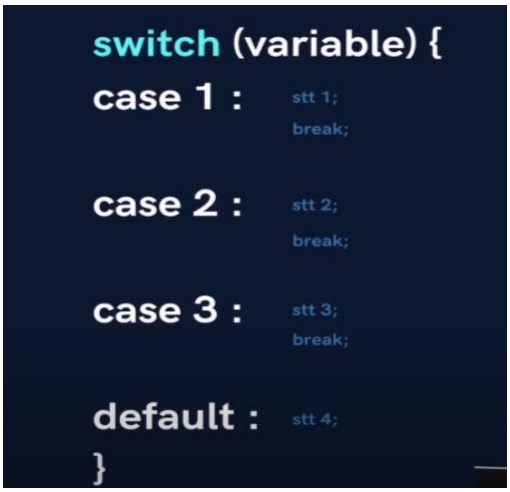
2. else-if

- Checks multiple conditions sequentially.
-
```
if (marks >= 90) {
    System.out.println("Grade: A");
} else if (marks >= 75) {
    System.out.println("Grade: B");
} else {
```

```
    System.out.println("Grade: C");
}
```

- 

3. switch Statement



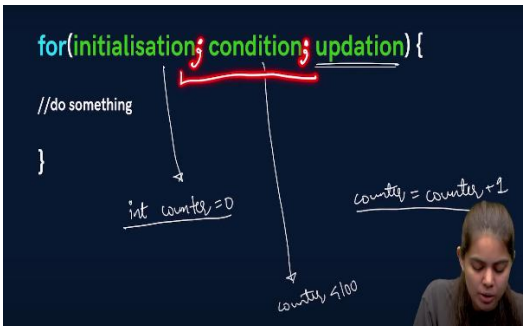
- Used when multiple conditions depend on a single variable.
- int day = 3;
  switch (day) {
  case 1: System.out.println("Monday"); break;
  case 2: System.out.println("Tuesday"); break;
  case 3: System.out.println("Wednesday"); break;
  default: System.out.println("Invalid day");
  }

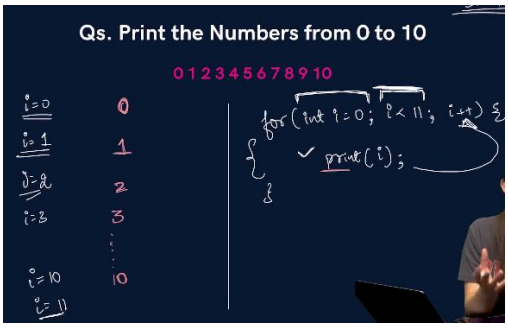
LOOPS

- A loop is used for executing a block of statements repeatedly until a particular condition is satisfied.
- A loop consists of an initialization statement, a test condition and an increment statement.)

Types of Loops :

1. For loop



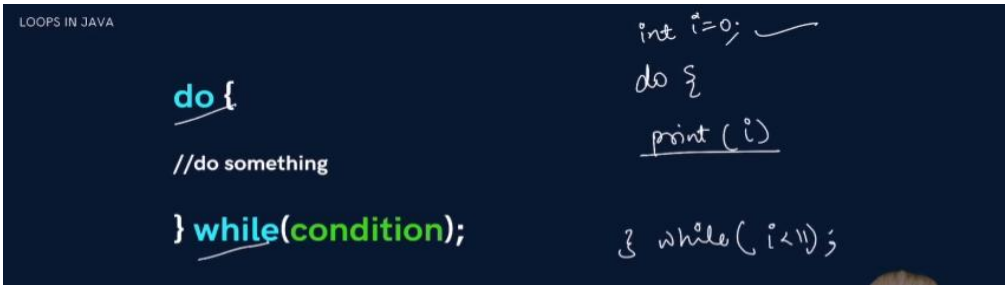


- Runs a block of code a specific number of times.
- for (int i = 1; i <= 5; i++) {
  System.out.println("Iteration: " + i);
  }

2. While loop

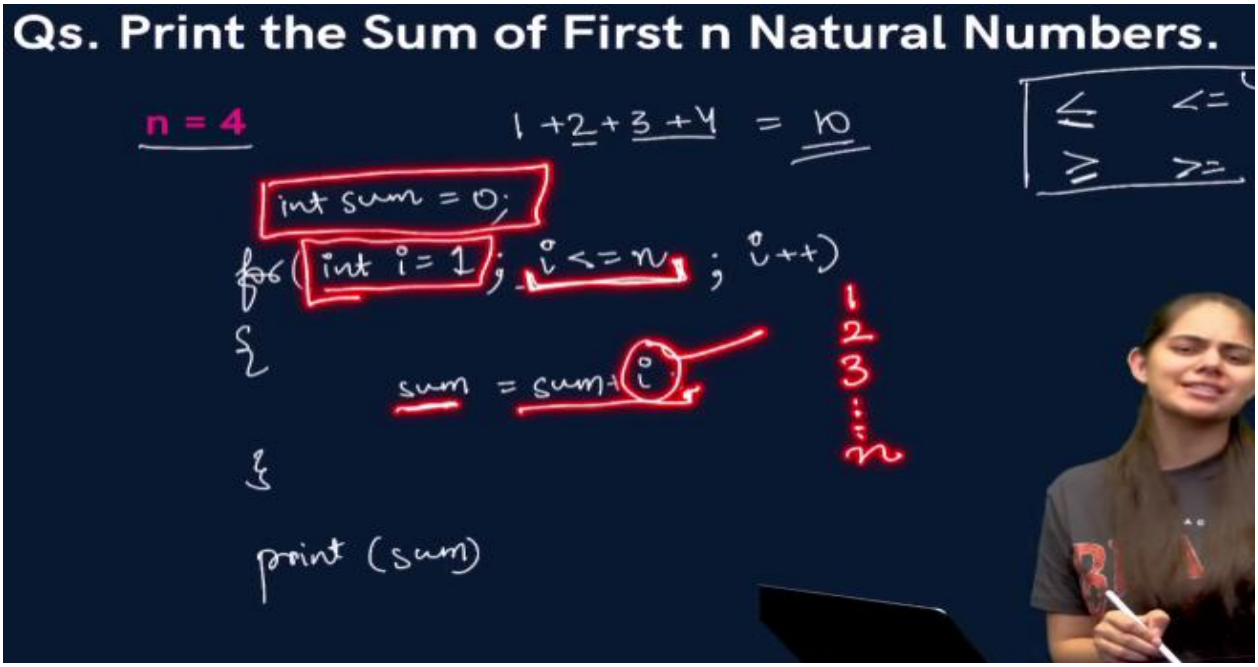
- Runs a block of code **while** a condition is true.
  int i = 1;
  while (i <= 5) {
  System.out.println("Iteration: " + i);
  i++;
  }

3. Do while loop



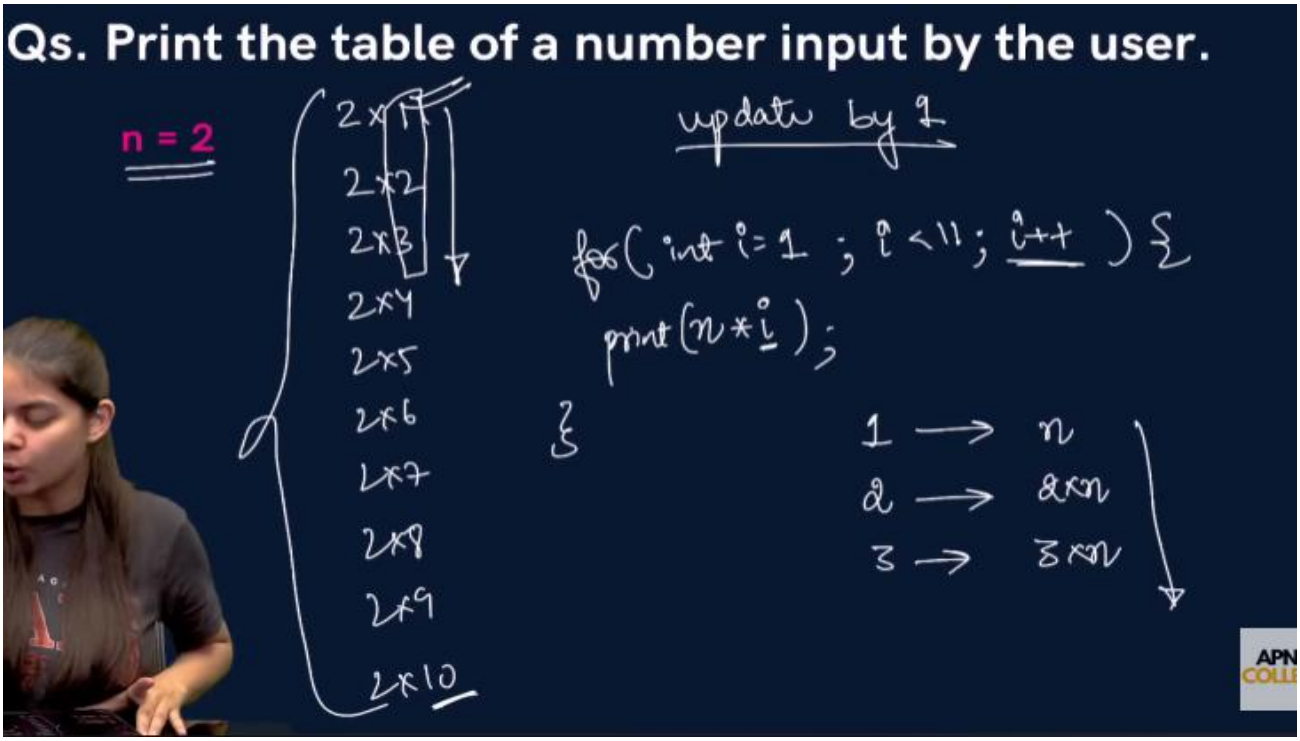
- Runs the code **at least once**, then repeats **while** the condition is true.
- int i = 1;
  do {
  System.out.println("Iteration: " + i);
  i++;
  } while (i <= 5);

Egs on Loops




```java
package Loops;

import java.util.*;

public class eg {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no :");
        int n = sc.nextInt();
        int sum = 0;

        for (int i = 1; i <= n; i++) {
            sum = sum + i;
        }
        System.out.println("The sum of the no is : " + sum);
        sc.close();
    }
}
```

```
Enter the no :
10
The sum of the no is : 55
PS D:\JAVA>
```

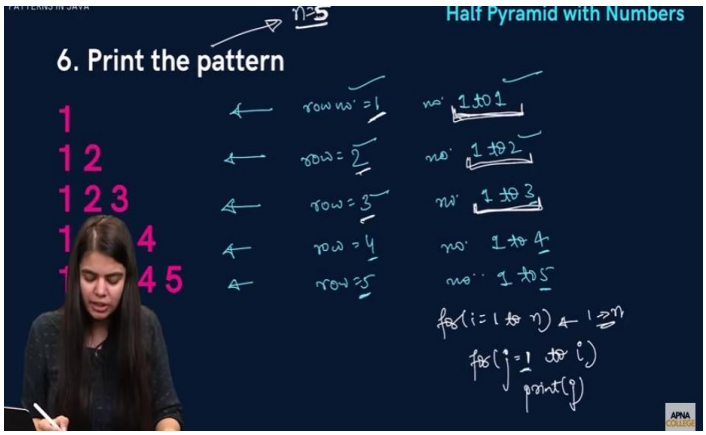



```java
package Loops;

import java.util.Scanner;

public class eg2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no :");
        int n = sc.nextInt();

        for (int i = 1; i < 11; i++) {
            System.out.println(i * n);
        }
        sc.close();
    }
}
```

```
5
10
15
20
25
30
35
40
45
50
PS D:\JAVA>
```

PATTERNS

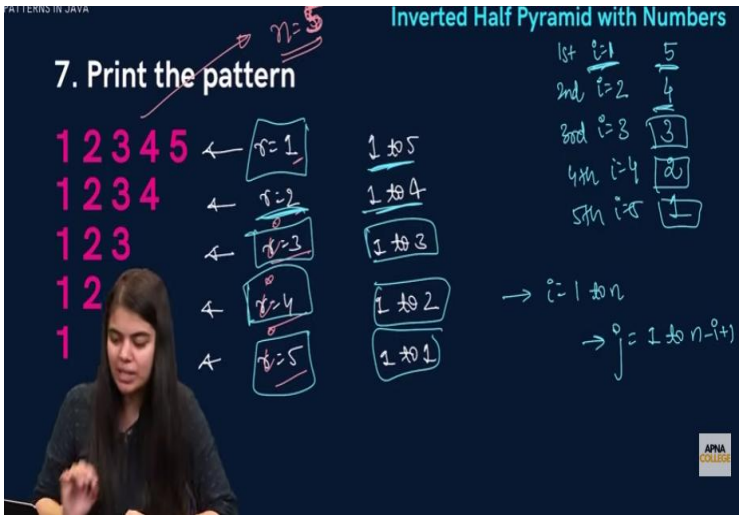
Half Pyramind with Numbers




```java
package Patterns;

import java.util.*;

public class HalfPyramidWithNumbers {
    Run | Debug
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.print(s: "Enter the no : ");
        int n = sc.nextInt();

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

        sc.close();
    }
}
```

```
Enter the no : 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

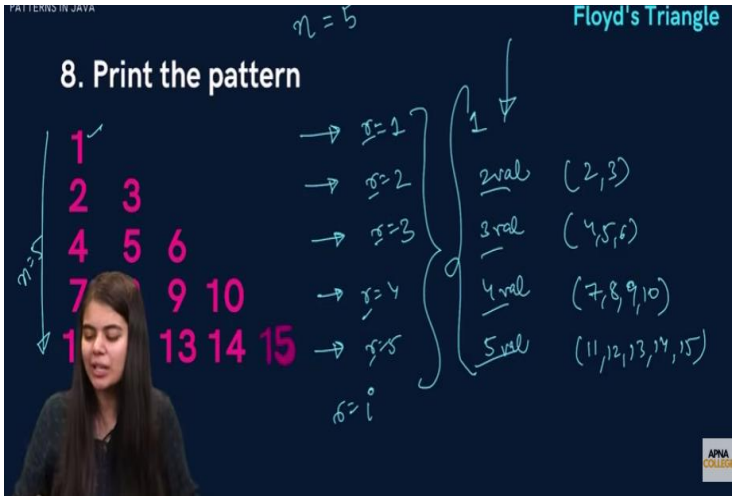
Inverted Half Pyramind with Numbers




```java
package Patterns;

import java.util.*;

public class InvertedHalfPyramidWithNumbers {
    Run | Debug
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.print(s: "Enter the no : ");
        int n = sc.nextInt();

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

        sc.close();

    }
}
```

```
Enter the no : 10
12345678910
123456789
12345678
1234567
123456
12345
1234
123
12
1
```

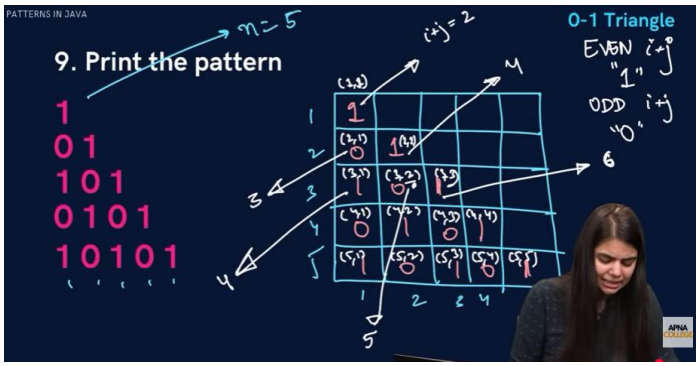
Floyd's Triangle

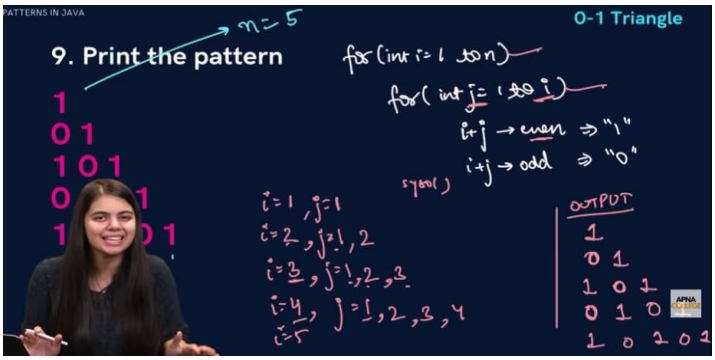



```java
package Patterns;

import java.util.*;

public class FloydsTrainagle {
    Run | Debug
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.print(s: "Enter the no : ");
        int n = sc.nextInt();
        int number = 1;

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

        sc.close();
        }
}
```

```
Enter the no : 5
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

0-1 Triangle

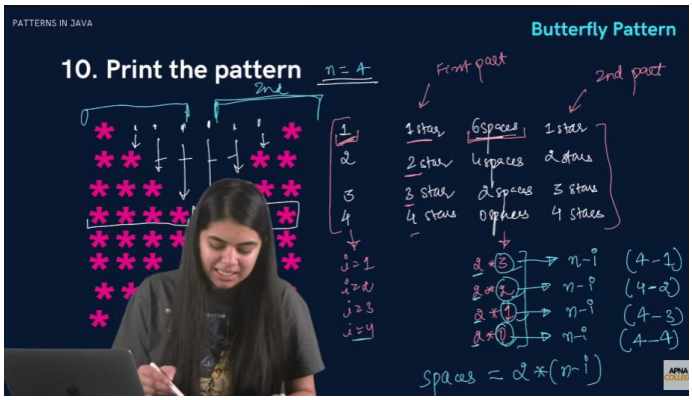


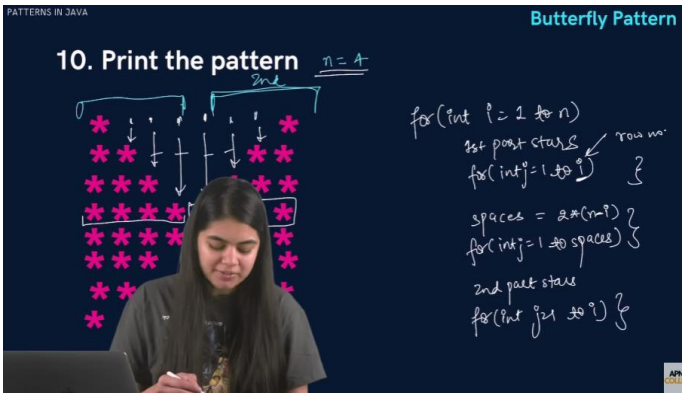


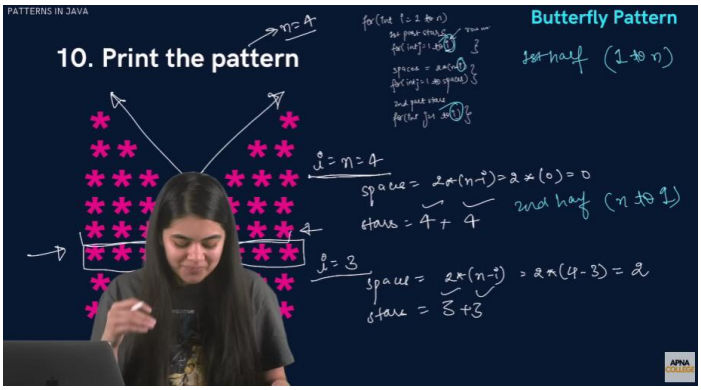

```java
package Patterns;

import java.util.Scanner;

public class ZeroOneTriangle {

    Run | Debug
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.print(s: "Enter the no : ");
        int n = sc.nextInt();

        //outer loop
        for(int i = 1; i <= n; i++){
            //inner loop
            for(int j=1 ; j <= i; j++){
                int sum = i+j;
                if(sum % 2 == 0){ //even value
                    System.out.print("1" + " ");
                } else { // odd value
                    System.out.print("0" + " ");
                }
            }
            System.out.println();
        }

        sc.close();
        }
}
```

```
Enter the no : 5
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

Butterfly Pattern

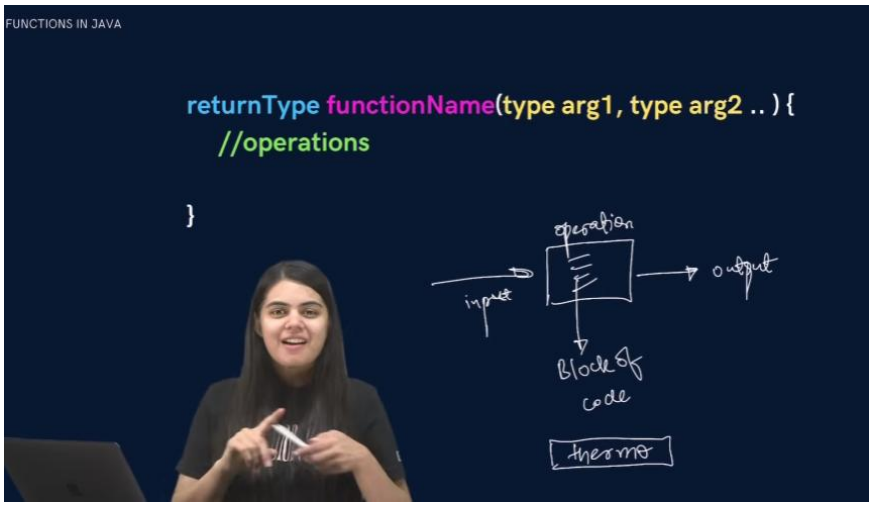







```java
package AdvancedPatterns;

public class ButterflyPattern {
    Run | Debug
    public static void main(String[] args) {
        int n=4;

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

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

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

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

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

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

}
```

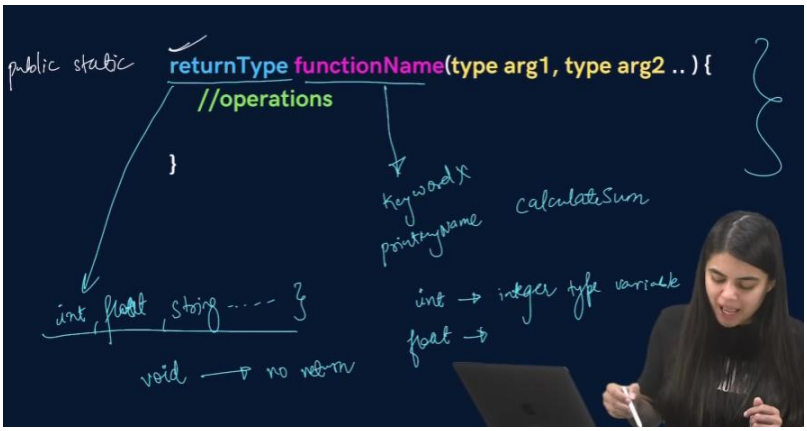
Functions



A function is a block of code that performs a specific task.

Why are functions used?

- If some functionality is performed at multiple places in software, then rather than writing the same code, again and again, we create a function and call it everywhere. This helps reduce code redundancy.
- Functions make maintenance of code easy as we have to change at one place if we make future changes to the functionality.
- Functions make the code more readable and easy to understand.



The **syntax** for function declaration is :

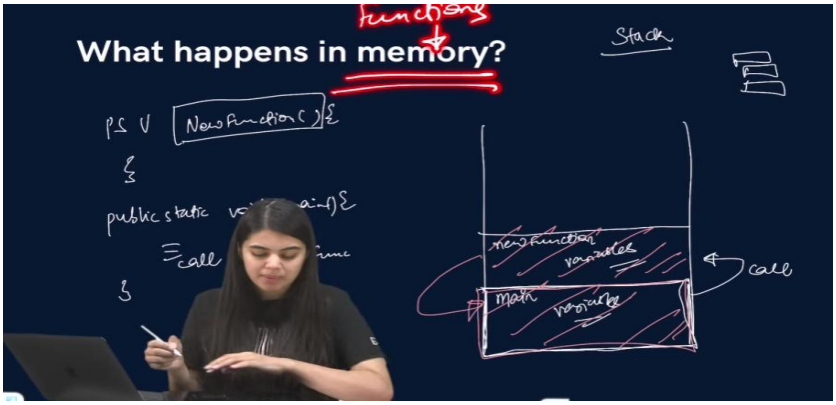
return-type function_name (parameter 1, parameter2, …… parameter n){

//function_body

}

return-type

The **return type** of a function is the data type of the variable that that function returns.



For eg - If we write a function that adds 2 integers and returns their sum then the return type of this function will be 'int' as we will return a sum that is an integer value.

When a function does not return any value, in that case the return type of the function is 'void'.

function_name

It is the unique name of that function.

It is always recommended to declare a function before it is used.

Parameters

A function can take some parameters as inputs. These parameters are specified along with their data types.

For eg- if we are writing a function to add 2 integers, the parameters would be passed like –

int add (int num1, int num2)

main function

The main function is a special function as the computer starts running the code from the beginning of the main function. Main function serves as the entry point for the program.

Example :

```
package Functions;

public class Main{
    public static void sum(int a, int b){
        int sum = a + b;
        System.out.println(sum);
    }
    public static void main(String args[]){
        int a = 10;
 int b = 20;
     sum(a, b); // Function Call
}}
```

Qs. Write a function to multiply 2 numbers.

```java
package Functions;

import java.util.Scanner;

public class Product {

    public static int calculateProduct(int a, int b){
        return a*b;
    }
    Run | Debug
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the 1st no : ");
        int a = sc.nextInt();

        System.out.print(s: "Enter the 2nd no : ");
        int b = sc.nextInt();

        int result = calculateProduct(a,b);

        System.out.println("Product of 2 nos is : "
        +result);

        sc.close();
    }
}
```

Qs. Write a function to calculate the factorial of a number

```java
package Functions;

import java.util.*;

public class Factorial {

    public static void printFactorial(int n){

        if(n<0){
            System.out.println(x: "Invalid Number");
            return;
        }
        int factorial = 1;
        //loop
        for(int i=n; i>=1; i--){
            factorial= factorial*i;
        }
        System.out.println(factorial);
        return;
    }
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the no : ");
        int n = sc.nextInt();

        printFactorial(n);

        sc.close();
    }
}
```

```
Enter the no : 5
120
```

1. **Make a function to check if a number is prime or not.**

```java
package Functions;

import java.util.*;

public class Prime {

    public static void checkPrime(int n){
        if(n <2){
            System.out.println(x: "Enter the no that
            should be Greater than or equal 2");
            return;
        }

        boolean isPrime= true;

        for(int i=2; i<=Math.sqrt(n); i++){

        if( n % i == 0){
            isPrime=false;
            break;
        }
        }

        if( isPrime){
            System.out.println(n + "Is is a Prime no");
        }else{
            System.out.println(n + "Is is not a Prime no");
        }
    }
```

```java
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the no : ");
        int n = sc.nextInt();

        checkPrime(n);

        sc.close();
    }

}
```

```
Enter the no : 2
2Is is a Prime no
PS D:\JAVA> ^C
PS D:\JAVA>
PS D:\JAVA>  d:; cd 'c
-hotspot\bin\java.exe'
SUS\AppData\Roaming\Co
dhat.java\jdt_ws\JAVA_
Enter the no : 4
4Is is not a Prime no
```

2. Make a function to check if a given number n is even or not.

```java
package Functions;

import java.util.*;

public class EvenOrOdd {

    public static void checkEven(int n){
        if(n<1){
            System.out.println(x: "Enter no greater than
            0");
        }else if(n % 2== 0){
            System.out.println(n+ " is an even no");
        }else{
            System.out.println(n+ " is Not an even no");
        }

        return ;
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);

        System.out.print(s: "Enter the no : ");
        int n= sc.nextInt();

        checkEven(n);
        sc.close();
    }
}
```

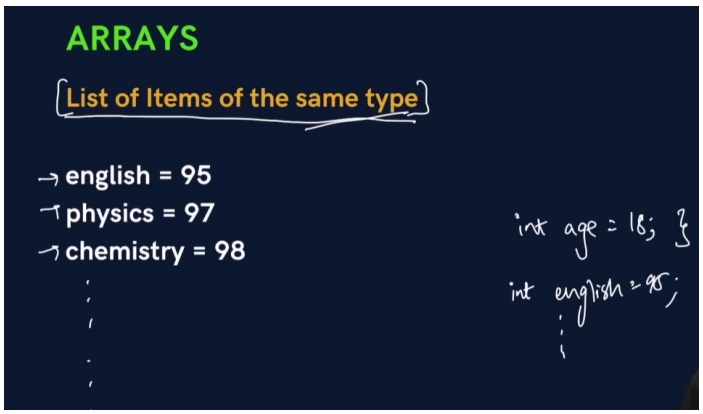
```
Enter the no : 39
39 is Not an even no
PS D:\JAVA> ^C
PS D:\JAVA>
PS D:\JAVA>  d:; cd
-hotspot\bin\java.exe
SUS\AppData\Roaming\C
dhat.java\jdt_ws\JAVA
Enter the no : 98
98 is an even no
```

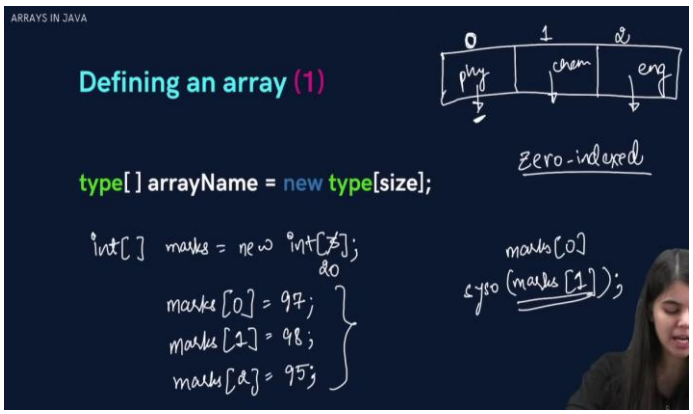
Make a function to print the table of a given number n.

```java
package Functions;

import java.util.*;

public class Table {

    public static void printMyTable(int n){
        for(int i =1; i<=10; i++){
            int result = n*i;
            System.out.println(n+ " * " +i+ " = " +result);
        }
        return;
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the no : ");
        int n = sc.nextInt();

        printMyTable(n);

        sc.close();
    }
}
```

```
Enter the no : 2
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
```

ARRAYS





1. Arrays are **objects**
2. Printing array name prints **address (hashcode)**
3. Use Arrays.toString() to print all elements
4. Array index starts from **0**
5. Stored in **heap memory**

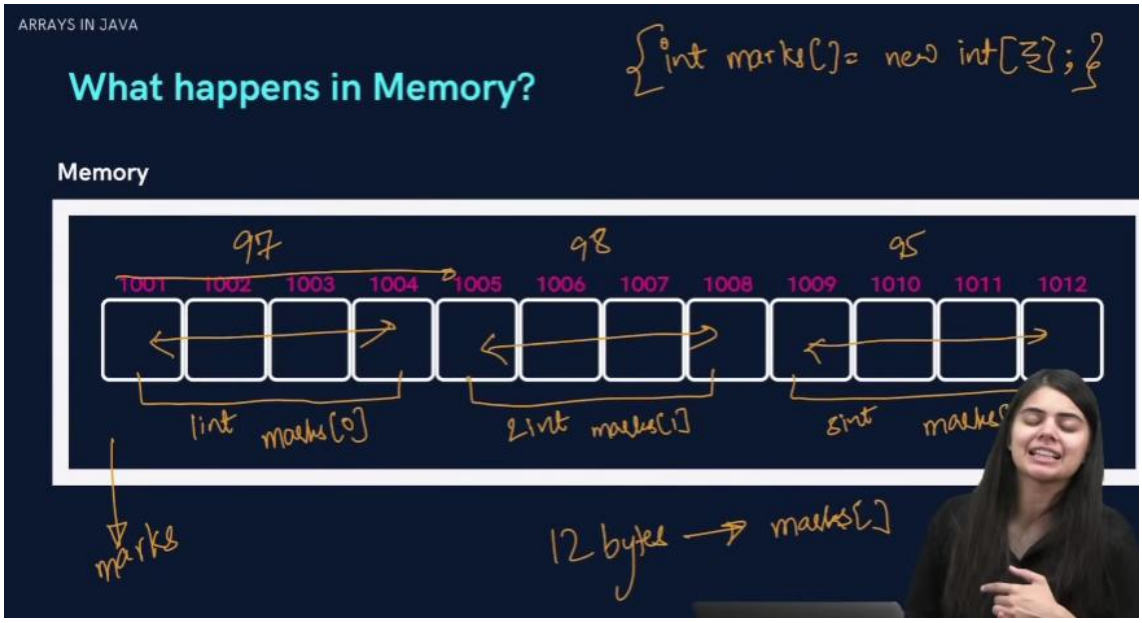
EXAMPLE.

```java
package Arrays;

public class FirstArray {
    Run | Debug
    public static void main(String[] args) {
        // int[] marks = new int[3]; this is also correct
        int marks[] = new int[3];
        marks[0] = 99;
        marks[1] = 97;
        marks[2] = 98;
        // System.out.println(marks[0]);
        // System.out.println(marks[1]);
        // System.out.println(marks);

        for(int i = 0; i <3; i++){
            System.out.println(marks[i]);
        }
    }
}
```

```
99
97
[I@2c7b84de
```

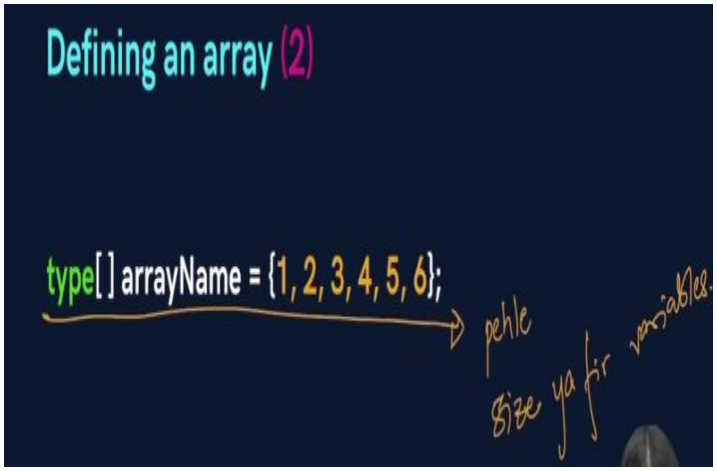
Command	Output	Represents
marks[0]	99	Element at index 0
marks[1]	97	Element at index 1
System.out.println(marks)	[I@hexCode	Array object reference

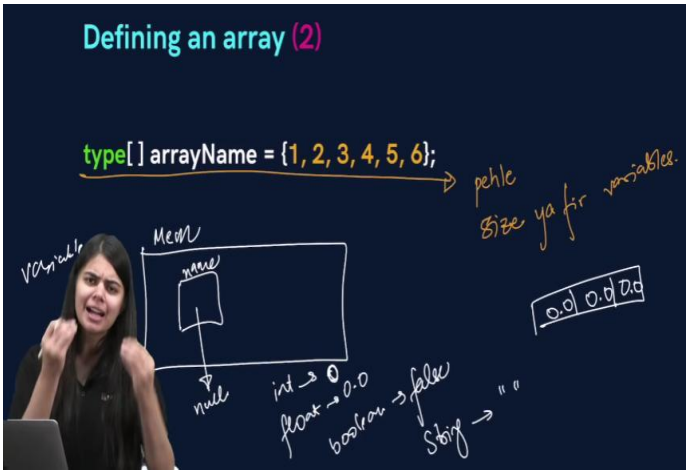
Part	Meaning
[	It's an **array**
I	Type of array → **int** (I stands for Integer type in JVM)
@	Separator
2c7b84de	**Hashcode / memory reference** of array




```java
package Arrays;

public class FirstArray {
    Run | Debug
    public static void main(String[] args) {
        // int[] marks = new int[3]; this is also correct
        // int marks[] = new int[3];
        int marks [] = {99,97,98};
        // marks[0] = 99;
        // marks[1] = 97;
        // marks[2] = 98;


        for(int i = 0; i <3; i++){
            System.out.println(marks[i]);
        }
    }
}
```

```
99
97
[I@2c7b84de
```





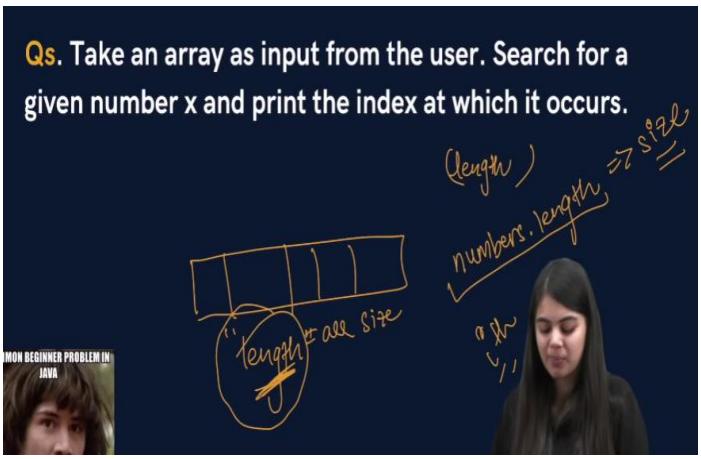
```java
package Arrays;

import java.util.*;

public class Arrays {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the size of the Array :
        ");
        int size = sc.nextInt();
        int numbers[] = new int[size];

        for(int i = 0; i < size; i++){
            System.out.println(numbers[i]);
        }

        sc.close();
    }
}
```

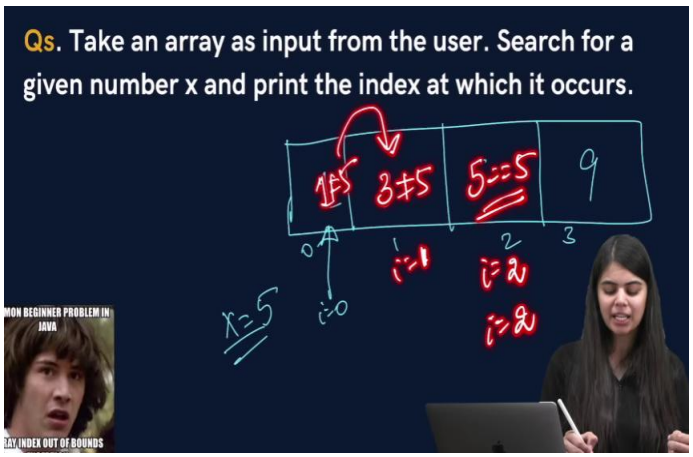
```
Enter the size of the Array : 5
0
0
0
0
0
```

```java
package Arrays;

import java.util.*;

public class Arrays {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the size of the Array :
        ");
        int size = sc.nextInt();
        int numbers[] = new int[size];

            //this loop is for taking the input from the
            users
        for(int i = 0; i < size; i++){
            numbers[i]= sc.nextInt();
        }

        System.out.print(s: "The numbers are : ");
        //this loop is for the output
        for(int i = 0; i < size; i++){
            System.out.print(numbers[i]+ " ");
        }

        sc.close();
    }
}
```

```
Enter the size of the Array : 5
1
2
3
4
5
The numbers are : 1 2 3 4 5
```

Linear Search






```java
package Arrays;

import java.util.*;

public class LinearSearch {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the size of the Array :
        ");
        int size = sc.nextInt();

        int numbers[] = new int[size];
        for(int i = 0; i < size; i++){
            numbers[i] = sc.nextInt();
        }

        System.out.print(s: "Enter the numbe to be
        searched : ");
        int x = sc.nextInt();

        for(int i = 0; i < numbers.length; i++){
            if(numbers[i] == x){
                System.out.println("X found at index : "
                +i);
            }

        }

        sc.close();
    }
}
```

```
Enter the size of the Array : 4
1
3
5
9
Enter the numbe to be searched : 5
X found at index : 2
```

```
Enter the size of the Array : 4
1
2
2
4
Enter the numbe to be searched : 2
X found at index : 1
X found at index : 2
```

1. **Take an array of names as input from the user and print them on the screen.**

```java
package Arrays;

import java.util.*;

public class ArrayNames {
    Run | Debug
    public static void main(String args[]){
        Scanner sc = new Scanner((System.in));

        System.out.print(s: "Enter the no of names to be
        printed : ");
        int size = sc.nextInt();

        String names [] = new String[size];
        for(int i = 0; i < size; i ++){
            names[i] = sc.next();
        }


        for(int i = 0; i < names.length; i++){
            System.out.println("name " + (i+1) +" is : "
            + names[i]);
        }

        sc.close();
    }
}
```

```
Enter the no of names to be printed : 3
Vinay
Vilas
Akerkar
name 1 is : Vinay
name 2 is : Vilas
name 3 is : Akerkar
```

2. **Find the maximum & minimum number in an array of integers.**

```java
package Arrays;

import java.util.*;

public class MaxMin {
    Run | Debug
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the no of Elemenets :
        ");
        int size = sc.nextInt();

        int numbers [] = new int[size];
        for(int i = 0; i < size; i++){
            numbers[i] = sc.nextInt();
        }

        int max = Integer.MIN_VALUE;
        int min = Integer.MAX_VALUE;

        for(int i = 0; i< numbers.length; i++){
            if(numbers[i] > max){
            max = numbers[i];
        }

        if(numbers[i] < min){
            min = numbers[i];
        }
        }

        System.out.println("Largest no is : "+max);
        System.out.println("Smallest no is : "+min);
        sc.close();
    }
}
```

```
Enter the no of Elemenets : 4
17
4
8
22
Largest no is : 22
Smallest no is : 4
```

3. **Take an array of numbers as input and check if it is an array sorted in ascending order**

Eg : { 1, 2, 4, 7 } is sorted in ascending order.

{3, 4, 6, 2} is not sorted in ascending order.

```java
package Arrays;


import java.util.*;

public class SortedAscendingOrder {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the No of Elements : ");
        int size = sc.nextInt();

        int numbers[] = new int[size];
        for(int i =0; i < size; i++){
            numbers[i] = sc.nextInt();
        }

        boolean isAscending = true;

        for(int i = 0; i < numbers.length-1; i++){
            if(numbers[i] > numbers[i + 1]){
                isAscending = false;
                break;
            }
        }

        if(isAscending){
            System.out.print(s: "The array is sorted in
            ascending order");
        }else{
            System.out.print(s: "The array is not sorted
            in ascending order");
        }
        sc.close();
    }
}
```

```
Enter the No of Elements : 4
1
2
3
4
The array is sorted in ascending order
```

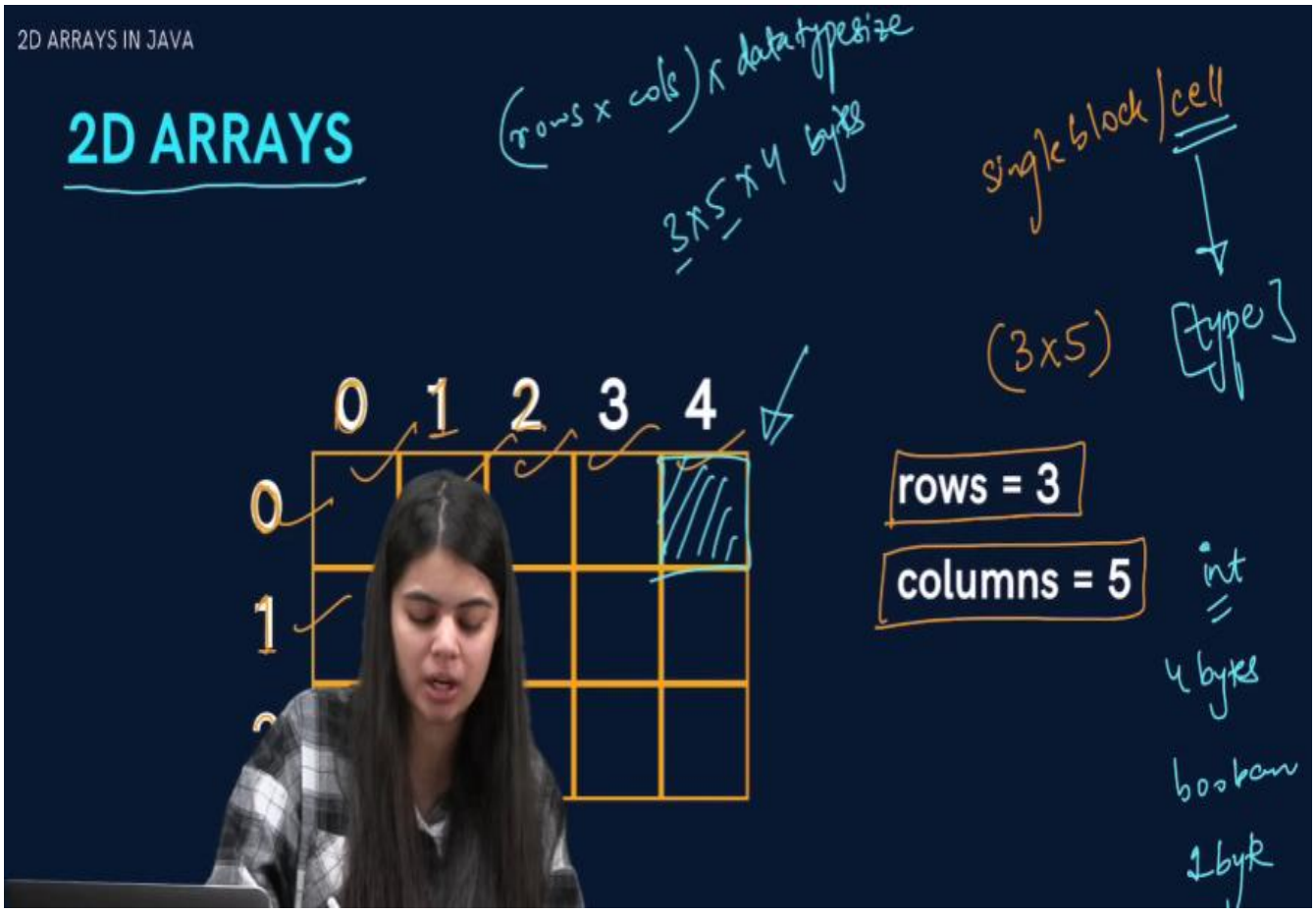
```
Enter the No of Elements : 4
3
2
5
1
The array is not sorted in ascending order
```

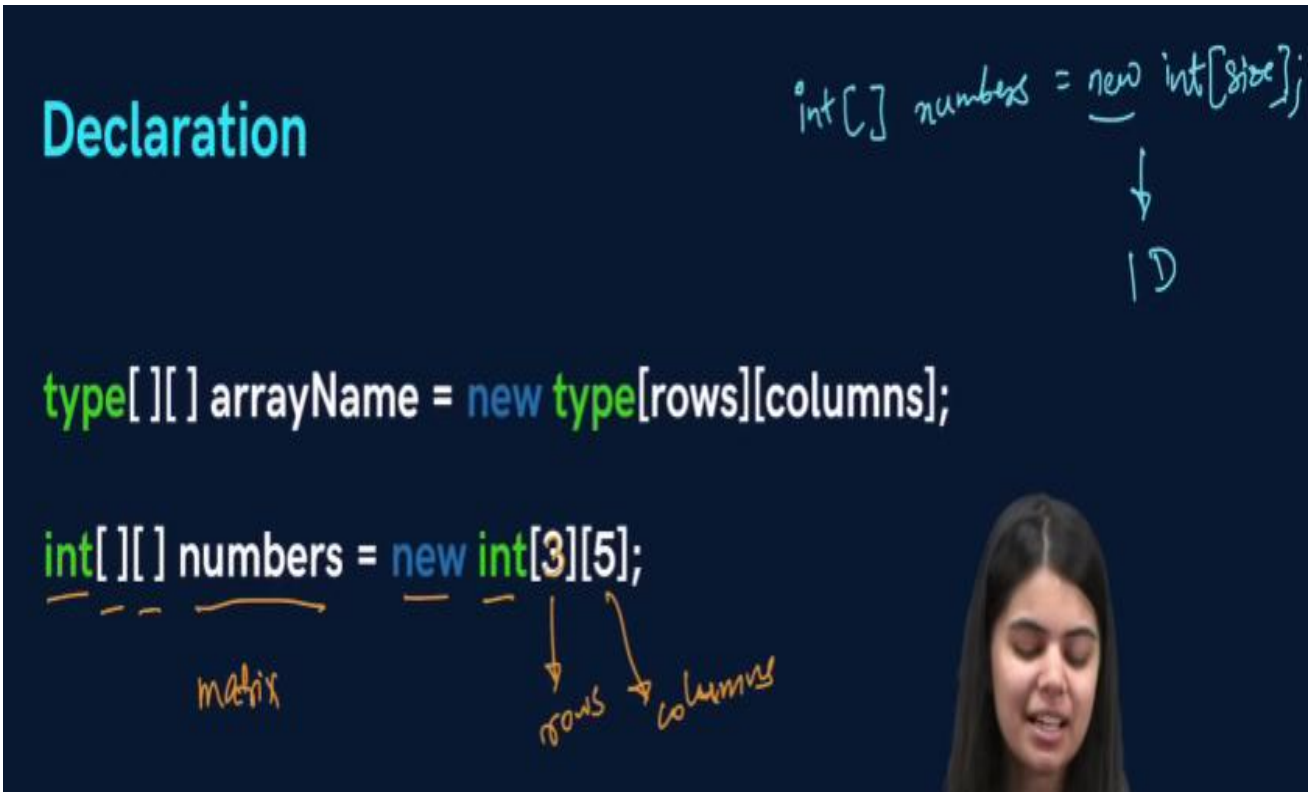
2D Arrays

It is similar to 2D matrices that we studied in 11th and 12th class.

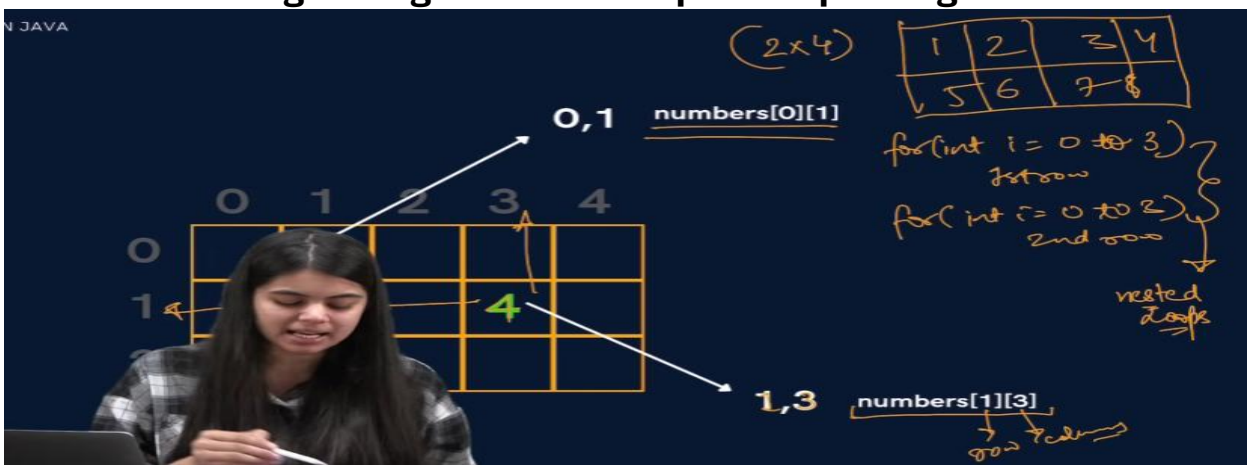
a. Creating a 2D Array - with new keyword

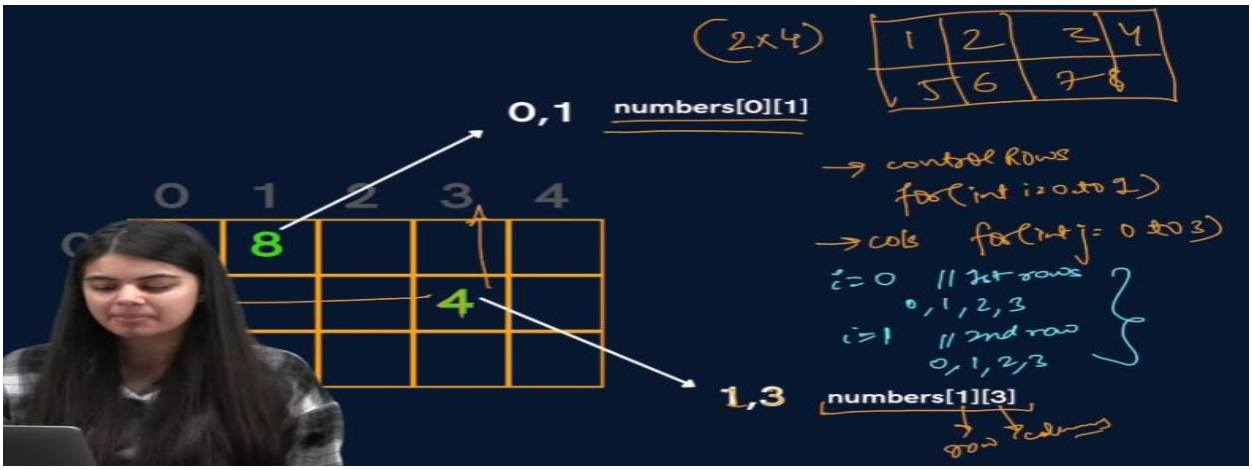
```
int[][] marks = new int[3][3];
```





b. Creating Taking a matrix as input and printing its elements.

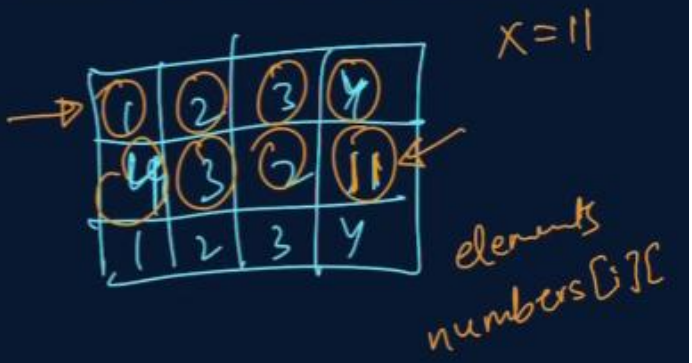





```java
import java.util.*;

public class TwoDArrays{
    Run | Debug
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the no of Rows : ");
        int rows = sc.nextInt();

        System.out.print(s: "Enter the no of Columns : ");
        int cols = sc.nextInt();

        int numbers[][] = new int[rows][cols];

        //input
        //rows
        for(int i = 0; i < rows; i++){
            //columns
            for(int j = 0; j < cols; j++){
                numbers[i][j] = sc.nextInt();
            }
        }

        //output
        for(int i = 0; i < rows; i++){
            //columns
            for(int j = 0; j < cols; j++){
                System.out.print(numbers[i][j]+ " ");
            }
            System.out.println();
        }
        sc.close();
    }
}
```

```
Enter the no of Rows : 3
Enter the no of Columns : 5
1 2 3 4 5
0 9 8 7 6
1 2 4 5 0
1 2 3 4 5
0 9 8 7 6
1 2 4 5 0
```

Qs. Take a matrix as input from the user. Search for a given number x and print the indices at which it occurs.




```java
import java.util.*;

public class MatrixSearch {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the no of Rows : ");
        int rows = sc.nextInt();

        System.out.print(s: "Enter the no of Columns : ");
        int cols = sc.nextInt();

        int numbers[][] = new int [rows][cols];

        for(int i = 0; i < rows; i++){
            for(int j = 0; j < cols; j++){
                numbers[i][j] = sc.nextInt();
            }
        }

        System.out.print(s: "enter the no to be Searched
        :  ");
        int x = sc.nextInt();

        for(int i = 0; i < rows; i++){
            for(int j = 0; j < cols; j++){
                if(numbers[i][j] == x){
                    System.out.print(" X is found at
                    location (" + i + ", " + j + ")");
                }
            }
        }
        sc.close();
    }
}
```

```
Enter the no of Rows : 3
Enter the no of Columns : 4
1 2 3 4
0 9 8 7
11 12 13 14
enter the no to be Searched :  11
 X is found at location (2, 0)
```

1. **For a given matrix of N x M, print its transpose.**

```java
import java.util.*;

public class MatrixTranspose {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the no of Rows : ");
        int rows = sc.nextInt();

        System.out.print(s: "Enter the no of Columns : ");
        int cols = sc.nextInt();

        int matrix [][] = new int[rows][cols];
        for(int i = 0; i < rows; i++){
            for(int j = 0; j < cols; j++){
                matrix[i][j] = sc.nextInt();
            }
        }

        System.out.println(x: "The transpose is : ");

        //To print transpose
        for(int j = 0; j < cols; j++){
            for(int i = 0; i < rows; i++){
                System.out.print(matrix[i][j]+ " ");
            }
            System.out.println();
        }
        sc.close();
    }
}
```

```
Enter the no of Rows : 2
Enter the no of Columns : 3
1 2 3
4 5 6
The transpose is :
1 4
2 5
3 6
```

Strings (STRINGS ARE IMMUTABLE)

Declaration

```java
String name = "Vinay";
```

Taking Input

```java
Scanner sc = new Scanner(System.in);
System.out.print(s: "Enter Your Name : ");
// String name = sc.next();
String name = sc.nextLine();
System.out.println("Your name is : " +name);
```

```
Enter Your Name : Vinay
Your name is : Vinay
```

Concatenation (Joining 2 strings)

```
String firstName = "Vinay";
String lastName = "Akerkar";
// String fullName = firstName + lastName;
String fullName = firstName + " " + lastName;
System.out.println(fullName);
```

```
Vinay Akerkar
```

Print length of a String

```
String firstName = "Vinay";
String lastName = "Akerkar";
// String fullName = firstName + lastName;
String fullName = firstName + " " + lastName;
System.out.println(fullName.length());
```

```
13
```

Access characters of a string

```
String firstName = "Vinay";
String lastName = "Akerkar";
// String fullName = firstName + lastName;
String fullName = firstName + " " + lastName;
System.out.println(fullName.length());

for(int i = 0; i < fullName.length(); i++){
    System.out.println(fullName.charAt(i));
}
```

```
13
V
i
n
a
y

A
k
e
r
k
a
r
```

Compare 2 strings

```
package Strings;

public class CompareTo {
    Run | Debug
    public static void main(String[] args) {
        String name1 = "Vinay";
        String name2 = "Vinay";

            //1. s1 > s2 : +ve value
            //2. s1 = s2 : 0
            //3. s1 < s2 : -ve value

        if(name1.compareTo(name2) == 0){
            System.out.println(x: "Strings are equal");
        }else {
            System.out.println(x: "Strings are NOT equal");
        }

        //DO NOT USE == to check for string equality
       //Gives correct answer here
        if(name1 == name2){
            System.out.println(x: "Strings are equal");
        }else {
            System.out.println(x: "Strings are NOT equal");
        }

            //Gives incorrect answer here
         if(new String(original: "Vinay") == new String
         (original: "Vinay")){
            System.out.println(x: "Strings are equal");
        }else {
            System.out.println(x: "Strings are NOT equal");
        }
    }
}
```

```
Strings are equal
Strings are equal
Strings are NOT equal
```

Substring

The substring of a string is a subpart of it

```
    String sentence = "VinayAkerkar";
    String name = sentence.substring(beginIndex: 0
,endIndex: 5);
    System.out.println(name);
}
```

```
Vinay
```

ParseInt Method of Integer class

```
public class ParseInt {
    Run | Debug
    public static void main(String[] args) {
        String str = "123";
        int number = Integer.parseInt(str);
        System.out.println(number);
    }
}
```

```
123
```

ToString Method of String class

```
public class ToString {
    Run | Debug
    public static void main(String[] args) {
        int number = 123;
        String str = Integer.toString(number);
        System.out.println(str.length());
    }
}
```

```
3
```

1. **Take an array of Strings input from the user & find the cumulative (combined) length of all those strings.**

```java
package Strings;

import java.util.*;

public class TotalStringLength {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the number of words
        whose combined Length you want : ");
        int size = sc.nextInt();
        int totLength = 0;

        String array[] = new String[size];
        for(int i = 0; i < size; i++){
            array[i] = sc.next();
            totLength += array[i].length();
        }

        System.out.print("The combined Length of the words
        is :"+totLength);
        sc.close();

    }
}
```

```
Enter the number of words whose combined Length you want : 3
vinay vilas akerkar
The combined Length of the words is :17
```

2. **Input a string from the user. Create a new string called ‘result’ in which you will replace the letter ‘e’ in the original string with letter ‘i’.**
   Example :
   original = “eabcdef’ ; result = “iabcdif”
   Original = “xyz” ; result = “xyz”

```java
package Strings;

import java.util.*;

public class ReplaceEWithI {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s: "Enter the String : ");
        String str = sc.next();

        String result = "";

        for(int i = 0; i < str.length(); i++){
            if(str.charAt(i) == 'e'){
                result += 'i';
            }else{
                result += str.charAt(i);
            }
        }
        System.out.println(result);

        sc.close();
    }
}
```

Iteration walkthroughs

Example A — `str = "eabcdef"` (user input)

Initial: `result = ""`

Index by index:

- i = 0 → `str.charAt(0) = 'e'` → branch true → `result = "" + 'i'` → `result = "i"`
- i = 1 → `str.charAt(1) = 'a'` → else → `result = "i" + 'a'` → `result = "ia"`
- i = 2 → `str.charAt(2) = 'b'` → else → `result = "iab"`
- i = 3 → `str.charAt(3) = 'c'` → else → `result = "iabc"`
- i = 4 → `str.charAt(4) = 'd'` → else → `result = "iabcd"`
- i = 5 → `str.charAt(5) = 'e'` → true → `result = "iabcd" + 'i'` → `result = "iabcd i"` → `"iabcd i"` → actually `"iabcd i"` without space → final `"iabcd i"`; combined: `result = "iabcd i"` → remove spacing: `"iabcd i"` → (final correct string is `"iabcdif"`)
  (Concise: `result` becomes `"iabcd i"` then next step:)
- i = 6 → `str.charAt(6) = 'f'` → else → `result = "iabcdif"`

```
Enter the String : eabcdef
iabcdif
```

3. Input an email from the user. You have to create a username from the email by deleting the part that comes after '@'. Display that username to the user.

   Example :

   email = "apnaCollegeJava@gmail.com" ; username = "apnaCollegeJava"

   email = "helloWorld123@gmail.com"; username = "helloWorld123"

```java
package Strings;

import java.util.*;

public class ExtractUsername {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s: "Enter the email : ");
        String email = sc.next();
        String username = "";

        for(int i = 0; i < email.length(); i++){
            if(email.charAt(i) == '@'){
                break;
            }else{
                username += email.charAt(i);
            }
        }
        System.out.println("The Extracted Username is : "
        +username);

        sc.close();
    }
}
```

Example 2 — `email = "helloWorld123@gmail.com"`

Initial:

- `email = "helloWorld123@gmail.com"`
- `userName = ""`

Per character:

- i = 0 → 'h' → `userName = "h"`
- i = 1 → 'e' → `userName = "he"`
- i = 2 → 'l' → `userName = "hel"`
- i = 3 → 'l' → `userName = "hell"`
- i = 4 → 'o' → `userName = "hello"`
- i = 5 → 'W' → `userName = "helloW"`
- i = 6 → 'o' → `userName = "helloWo"`
- i = 7 → 'r' → `userName = "helloWor"`
- i = 8 → 'l' → `userName = "helloWorl"`
- i = 9 → 'd' → `userName = "helloWorld"`
- i = 10 → '1' → `userName = "helloWorld1"`
- i = 11 → '2' → `userName = "helloWorld12"`
- i = 12 → '3' → `userName = "helloWorld123"`
- i = 13 → '@' → `break`

Final: `userName = "helloWorld123"` → printed.

```
Enter the email : helloWorld123@gmail.com
The Extracted Username is : helloWorld123
```