

JAVA

Contents

1. Introduction
2. Syntax
3. Variables & Data types
4. Taking input from user (Scanner class)
5. Operators
6. String
7. If-else
8. Switch case
9. Loops (for, while, do while)
10. Break & continue
11. Function
12. Recursion
13. Arrays
14. Multidimensional arrays

Java is a popular programming language, created in 1995.

It is owned by Oracle, and more than 3 billion devices run Java.

It is used for:

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

Syntax

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

- **public class Main** - Starting point of a java program, from here only the java program starts executing.
- **public static void main** - Here the word '**public**' is referred to as an access modifier means when it is public you can use the method wherever from the program. Now the word '**static**' helps to use functions without creating objects. And the word '**void**' means the function doesn't have a return value.
'**String[] args**' means the string arguments which the program will obtain.
- The **System.out.println();** is a function used to print data on the screen.
- '**println**' adds a new line while only '**print**' doesn't add a new line.

Naming Conventions

- For classes we use PascalConvention.
- For functions we use camelCaseConvention.

We use naming conventions because we can't use spaces in between functions, classes and variables.

NOTE

The name of the java file must match the class name. When saving the file, save it using the class name and add ".java" to the end of the filename.

Variables

Variables are containers for storing data values.

In Java, there are different types of variables, for example:

- **String** - stores text, such as "Hello". String values are surrounded by double quotes.
- **int** - stores integers (whole numbers), without decimals, such as 123 or -123.
- **float** - stores floating point numbers, with decimals, such as 19.99 or -19.99.
- **char** - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes.
- **boolean** - stores values with two states: true or false.

Data Types

- **Primitive** - int, float, bool, char, long, double, byte, short.
- **Non-Primitive** -

```
public class Main {  
    public static void main(String[] args) {  
  
        int num1 = 5;  
        int num2 = 10;  
        int num3 = 10;  
        int sum = num1 + num2 + num3;  
  
        System.out.println(sum);  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
  
        int num1 = 5;  
        float num2 = 5.9f; // Here we used f at the end of the numbers to define that this is a floating  
point value  
        double num3 = 7.997;  
        char ch = 'A';  
        byte num4 = 34;  
        boolean a = true;  
        short num5 = 25;  
        long num6 = 657888888;  
        String name = "Sudipto";  
  
        System.out.println("The value of int is : " + num1);  
    }  
}
```

```
System.out.println("The value of float is : " +num2);
System.out.println("The value of double is : " +num3);
System.out.println("The value of char is : " +ch);
System.out.println("The value of byte is : " +num4);
System.out.println("The value of boolean is : " +a);
System.out.println("The value of short is : " +num5);
System.out.println("The value of long is : " +num6);
System.out.println("The value of string is : " +name);
}
}
```

Taking Input From User

To take input from user we can use **scanner** class in java.

Here we imported a java file (**java.util.Scanner**) to use the scan function.

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        System.out.println("Taking input form user");
        Scanner sc = new Scanner(System.in);    // Here System.in means telling the computer to read
from the keyboard
        System.out.print("Enter no. 1 : ");
        int a = sc.nextInt();
        System.out.print("Enter no. 2 : ");
        int b = sc.nextInt();

        int sum = a + b;
        System.out.println("The sum of number 1 and 2 is :" +sum);

    }
}
```

Here we wrote a program to check whether an input is stored or not.

```
import java.util.Scanner;

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

        Scanner sc = new Scanner(System.in);    // Here System.in means telling the computer to read
from the keyboard
        System.out.print("Enter no. : ");

        boolean b1 = sc.hasNextInt();
        System.out.println(b1);

    }
}
```

Here we wrote a program to print a string

```
import java.util.Scanner;

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

        System.out.println("Enter a string");
        Scanner sc = new Scanner(System.in);
        String str = sc.next();
        System.out.println(str);

    }
}
```

Output :

```
Enter a string
New york city

New
```

Here we saw that the output line has only come with one word and it ignored the other words, so to print the entire line we can use **nextLine()** method.

```
import java.util.Scanner;

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

        System.out.println("Enter a string ");
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        System.out.println(str);

    }
}
```

Here's a more detail explanation.

```
import java.util.Scanner;

public class ScannerExample {
    public static void main(String[] args) {
        // Create a Scanner object to read from the console (System.in)
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter some input
        System.out.print("Enter your name: ");
```

```

// Read a line of text from the console and store it in a variable
String name = scanner.nextLine();

// Prompt the user to enter an integer
System.out.print("Enter your age: ");

// Read an integer from the console and store it in a variable
int age = scanner.nextInt();

// Display the entered values
System.out.println("Hello, " + name + "! You are " + age + " years old.");

// Close the scanner when you're done with it (optional but recommended)
scanner.close();
}
}

```

- **import java.util.Scanner;**:: Import the Scanner class from the **java.util** package.
- **public class ScannerExample {**:: Define a class named ScannerExample.
- **public static void main(String[] args) {**:: Start the main method, which is the entry point of the program.
- **Scanner scanner = new Scanner(System.in);**:: Create a new Scanner object named scanner that reads input from the console (**System.in**).
- **System.out.print("Enter your name: ");**:: Print a prompt asking the user to enter their name.
- **String name = scanner.nextLine();**:: Read a line of text (including spaces) from the console and store it in the variable name.
- **System.out.print("Enter your age: ");**:: Print a prompt asking the user to enter their age.
- **int age = scanner.nextInt();**:: Read an integer from the console and store it in the variable age.
- **System.out.println("Hello, " + name + "! You are " + age + " years old.");**:: Display a message using the values entered by the user.
- **scanner.close();**:: Close the Scanner object to release system resources (optional but recommended when you're done with it).

Exercise 1

Here is a program to calculate marks and percentage of a student of in 5 subjects out of 100 each.

```
import java.util.Scanner;

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

        Scanner sc = new Scanner(System.in);

        int Full_Marks = 100;

        System.out.print("Enter marks of computer : ");
        int Computer = sc.nextInt();
        System.out.print("Enter marks of science : ");
        int Science = sc.nextInt();
        System.out.print("Enter marks of english : ");
        int English = sc.nextInt();
        System.out.print("Enter marks of history : ");
        int History = sc.nextInt();
        System.out.print("Enter marks of geography : ");
        int Geography = sc.nextInt();

        int Total_Marks = (History + Geography + English + Computer + Science);
        int Percentage =(Total_Marks*Full_Marks/500);
        System.out.println("Your Total Marks is : " +Total_Marks);
        System.out.println("Your Percentage is : " +Percentage +"%");

    }
}
```

Operators

Operators are used to perform operations on variables and values.

Arithmetic Operators

+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
%	Modulus	$x \% y$
++	Increment	$++x$
--	Decrement	$--x$

Assignment Operators

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3

Comparison Operators

==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Logical Operators

Operator	Name	Description	Example
&&	Logical and	Returns true if both statements are true	x < 5 && x < 10
	Logical or	Returns true if one of the statements is true	x < 5 x < 4
!	Logical not	Reverse the result, returns false if the result is true	!(x < 5 && x < 10)

String

A string is a sequence of characters.

```
public class Main {  
    public static void main(String[] args) {  
  
        String name = new String("New York City ");  
        System.out.println(name);  
  
    }  
}
```

Or,

```
public class Main {  
    public static void main(String[] args) {  
  
        String name = "New York City";  
        System.out.println(name);  
  
    }  
}
```


String Methods

```
public class Main {  
    public static void main(String[] args) {  
  
        String name = "Sudipto";  
        System.out.println(name.length());  
  
    }  
}
```

- name.length() - Returns the length of the string.
- name.toLowerCase() - Returns the string in complete lower case.
- name.toUpperCase() - Returns the string in complete upper case.
- Name.replace() - Replaces a character with another one.

```
public class Main {  
    public static void main(String[] args) {  
  
        String name = "Sudipto";  
        System.out.println(name);  
        System.out.println(name.replace('o', 'a'));  
  
    }  
}
```

If-else

```
import java.util.Scanner;  
  
public class IfElse {  
    public static void main (String[] args){  
  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Enter a number : ");  
        int num = sc.nextInt();  
  
        if(num > 18){  
            System.out.println("Congrats! You are an adult now.");  
        }  
        else{  
            System.out.println("Sorry! You are still underage");  
        }  
  
    }  
}
```

```

import java.util.Scanner;

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

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter your marks to check grade : ");
        float marks = sc.nextFloat();

        if(marks >= 75 && marks <= 100){
            System.out.println("You got grade A ");
        }
        else if(marks >= 50 && marks <= 75){
            System.out.println("You got grade B ");
        }
        else if(marks >= 25 && marks <= 50){
            System.out.println("You got grade C ");
        }
        if(marks >= 0 && marks <= 25){
            System.out.println("You got grade D ");
        }
        else{
            System.out.println("Please enter valid marks.");
        }

    }
}

```

Switch Case

Enhanced switch case

```

import java.util.Scanner;

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

        System.out.print("Enter the number of day : ");
        int days = sc.nextInt();

        switch (days){
            case 1 -> System.out.println("Monday");
            case 2 -> System.out.println("Tuesday");
            case 3 -> System.out.println("Wednesday");
            case 4 -> System.out.println("Thursday");
            case 5 -> System.out.println("Friday");
            case 6 -> System.out.println("Saturday");
            case 7 -> System.out.println("Sunday");
        }
    }
}

```

```
default -> System.out.println("Please enter a valid number between 1 - 7 ");
}

}

}
```

For loop

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

        for (int i = 1; i <= 10; i++) {
            System.out.println(i*2);
        }

    }
}
```

Write a program to print the table of 2

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

        for(int i = 1 ; i <= 10 ; i ++){
            System.out.println("2 x " +i + " = " +i*2);
        }

    }
}
```

Write a program to take input of a number from user and print it's table

```
Import java.util.Scanner;

Public class Main{
    Public static void main(String[] args){
        Scanner sc = new Scanner(System.in);

        Int i;
        System.out.print("Enter the number you want to print table of:");
        Int num=sc.nextInt();

        for(i=1; i<=10; i++){
            System.out.println(num + " x " +i + " = " +num*i);
        }

    }
}
```

Taking Input From User

```
import java.util.Scanner;
public class Scan {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);    // Here System.in means telling the
computer to read from the keyboard
        System.out.println("Taking input form user");
        System.out.print("Enter no. 1 : ");
        int a = sc.nextInt();
        System.out.print("Enter no. 2 : ");
        int b = sc.nextInt();
        int sum = a + b;
        System.out.println("The sum of number 1 and 2 is : " +sum);
    }
}
```

If-else

```
import java.util.Scanner;
public class IfElse {
    public static void main (String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number : ");
        int num = sc.nextInt();
        if(num > 18){
            System.out.println("Congrats! You are an adult now.");
        }
        else{
            System.out.println("Sorry! You are still underage");
        }
    }
}
```

```
import java.util.Scanner;
public class IfElse2 {
    public static void main (String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your marks to check grade : ");
        float marks = sc.nextFloat();
        if(marks >= 75 && marks <= 100){
            System.out.println("You got grade A ");
        }
        else if(marks >= 50 && marks <= 75){
            System.out.println("You got grade B ");
        }
        else if(marks >= 25 && marks <= 50){
            System.out.println("You got grade C ");
        }
        if(marks >= 0 && marks <= 25){
```

```

        System.out.println("You got grade D ");
    }
    else{
        System.out.println("Please enter valid marks.");
    }
}
}

```

Switch Case

```

import java.util.Scanner;
public class SwitchCase {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of day : ");
        int days = sc.nextInt();
        switch (days){
            case 1 :
                System.out.println("Monday");
                break;
            case 2 :
                System.out.println("Tuesday");
                break;
            case 3 : System.out.println("Wednesday");
                break;
            case 4 : System.out.println("Thursday");
                break;
            case 5 : System.out.println("Friday");
                break;
            case 6 : System.out.println("Saturday");
                break;
            case 7 : System.out.println("Sunday");
                break;
            default : System.out.println("Please enter a valid number between 1 - 7 ");
                break;
        }
    }
}

```

Enhanced Switch Case

```

import java.util.Scanner;
public class EnhancedSwitchCase {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of day : ");
        int days = sc.nextInt();
        switch (days){
            case 1 -> System.out.println("Monday");
            case 2 -> System.out.println("Tuesday");
            case 3 -> System.out.println("Wednesday");
            case 4 -> System.out.println("Thursday");
            case 5 -> System.out.println("Friday");

```

```

        case 6 -> System.out.println("Saturday");
        case 7 -> System.out.println("Sunday");
        default -> System.out.println("Please enter a valid number between 1 - 7 ");
    }
}
}

```

For Loop

```

public class ForLoop {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            System.out.println(i*2);
        }
    }
}

public class ForLoop2 {
    public static void main(String[] args) {
        for(int i = 1 ; i <= 10 ; i++) {
            System.out.println("2 x " +i + " = " +i*2);
        }
    }
}

import java.util.Scanner;
public class ForLoop3{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int i;
        System.out.print("Enter the number you want to print table of:");
        int num=sc.nextInt();
        for(i=1; i<=10; i++){
            System.out.println(num + " x " +i + " = " +num*i);
        }
    }
}

```

While Loop

In the while loop the loop will execute until the given condition is true.

```

import java.util.*;
public class WhileLoop{
    public static void main(String args[]){
        int counter = 0;
        while(counter < 100){
            System.out.println("Muffin");
            counter++;
        }
    }
}

```

Do-While Loop

Do-while loop is quite similar to while loop but here's a change and that is in do-while loop first we do the work or execute the code and then check the given condition.

While loop - First condition check then execute the code.

Do-while loop - First execute the code then condition check.

```
import java.util.*;
public class DoWhile{
    public static void main(String args[]){
        int counter = 1;
        do{
            System.out.println("Muffin");
            counter++;
        }while(counter<=10);
    }
}
```

Break & Continue

BREAK: If the code reaches to a given condition at break then then the loop will end.

CONTINUE: Similarly, when the loop has given a continue statement then it skips the given condition or line and move on to the next statement.

```
import java.util.*;
public class Function{
    public static void main(String args[]){
        for(int i=1; i<=10; i++){
            if(i==5){
                break;
            }
            System.out.println(i);
        }
    }
}
```

// Output

1
2
3
4

Here in this, you can see that whenever the i became 5, the loop broke and the execution ends.

```
import java.util.*;
public class Function{
    public static void main(String args[]){
```

```

        for(int i=1; i<=10; i++){
            if(i==5){
                continue;
            }
            System.out.println(i);
        }
    }
}

```

// Output

```

1
2
3
4
6
7
8
9
10

```

Here also you can see that whenever used the continue word, the loop didn't stopped executing but it skipped the given condition number which is of number 5.

Function

Functions are kind of shortcuts to perform a certain task everytime we need it to without any extra implementation.

For example : as you can see here we have declared a function named 'printMuffin' which will return or print a word muffin every time we call it so in the main function when we called it it prints the word muffin.

```

public class Function{
    public static void printMuffin(){
        System.out.println("Muffin");
    }

    public static void main(String args[]){
        printMuffin(); // Function call
    }
}

```

Function with parameters

Here you can see that we have taken input from the user and printed the sum of a & b.

```

import java.util.*;
public class Function2{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the value of a : ");
        int a = sc.nextInt();
        System.out.print("Enter the value of b : ");
    }
}

```



```

        int b = sc.nextInt();
        int sum = a + b;
        System.out.println("Sum of a + b is : "+sum);
    }
}

```

But instead of this we can also write it as:

OPTION 1:

```

import java.util.*;
public class Function2{
    public static int sum(){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the value of a : ");
        int a = sc.nextInt();
        System.out.print("Enter the value of b : ");
        int b = sc.nextInt();
        int sum = a + b;
        System.out.println("Sum of a + b is : "+sum);
        return sum;
    }
    public static void main(String args[]){
        sum();
    }
}

```

Here we have wrapped up all the code inside a function which makes the code reusable everytime we need to perform sum operation.

```

import java.util.*;
public class Function2{
    public static int add(int a, int b){ // Here int a & int b are
formal parameters
        int sum = a + b;
        return sum;
    }
    public static void main(String args[]){
        System.out.println("Sum of a + b is : "+add(5,20)); // Here
the passed value 5 & 20 are called arguments or actual parameters
    }
}

```

Here in this function we took two parameters or say, int value as input and then the function performed.

Formal parameters = containers for values.

Actual parameters = real values.

There are two types of methods :

1. User defined methods - The methods which programmers use by themselves.
2. Inbuilt methods - The methods which is built in java to use like math class which has a lot of inbuilt methods like pow(), sqrt(), max() and so on.

Function Overloading

Function overloading in Java refers to the ability to define multiple functions in the same class or in the same hierarchy with the same name but different parameters.

3. Function/Method Name:

The name of the function or method.

4. Parameters:

The input values that a function or method accepts.

5. Return Type:

The data type of the value that the function or method returns.

```
public class Calculator {  
  
    // Function with two int parameters  
    public int add(int a, int b) {  
        return a + b;  
    }  
  
    // Function with three double parameters  
    public double add(double a, double b, double c) {  
        return a + b + c;  
    }  
  
    // Function with two strings as parameters  
    public String concatenate(String str1, String str2) {  
        return str1 + str2;  
    }  
  
    // Function with an int and a double as parameters  
    public double add(int a, double b) {  
        return a + b;  
    }  
  
    // Function with a double and an int as parameters  
    public double add(double a, int b) {  
        return a + b;  
    }  
  
    // Function with no parameters  
    public void sayHello() {  
        System.out.println("Hello!");  
    }  
  
    // ... other functions with different parameter combinations  
}
```

In the Calculator class, there are multiple functions with the same name (add), but they differ in the number or types of parameters they accept. For example:

add(int a, int b) takes two integers.

add(double a, double b, double c) takes three doubles.

concatenate(String str1, String str2) takes two strings.

This allows you to use the same name for logically similar operations, making the code more readable and intuitive.

When you call a function, Java determines which version of the function to execute based on the number and types of arguments you provide.

Recursion

Recursion in Java refers to a programming concept where a method calls itself in order to solve a problem. In other words, a method in Java can invoke itself to perform a task.

Recursion is often used in situations where a problem can be broken down into smaller sub-problems of the same type.

```
import java.util.*;
public class NewFactorial{
    public static int factorial(int n){
        if(n==0){
            return 1;
        }
        else{
            return n * factorial(n-1);
        }
    }
    public static void main(String args[]){
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number to find factorial : ");
        int number = scanner.nextInt();
        scanner.close();

        if(number<=0){
            System.out.println("Enter a non negative number.");
        }
        else{
            int result= factorial(number);
            System.out.println("Factorial of " +number +" is "
+factorial(number));
        }
    }
}
```

Arrays

an array is a data structure that allows you to store multiple values of the same type under a single variable name.

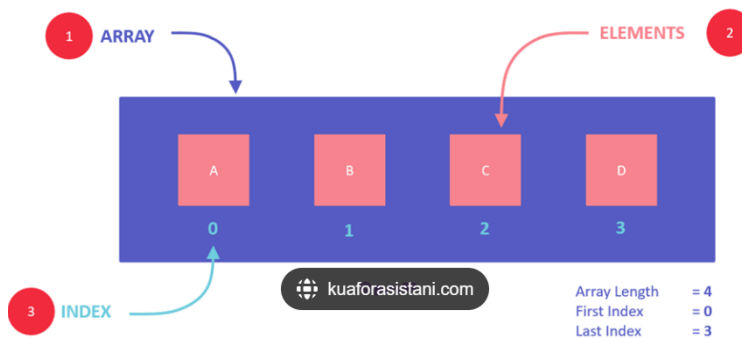
It is a container object that holds a fixed number of elements of the same data type.

Arrays are widely used in programming to represent collections of data, such as a list of numbers, a group of characters, or any other type of data.

6. Fixed Size: Once an array is created, its size (the number of elements it can hold) is fixed and cannot be changed.
7. Same Data Type: All elements in an array must be of the same data type. For example, if you declare an array of integers, all elements in that array must be integers.
8. Zero-Based Indexing: Array elements are accessed using an index, and the indexing starts from 0. The first element is accessed with index 0, the second with index 1, and so on.

```
public class ArrayExample {  
    public static void main(String[] args) {  
        // Declare and initialize an array of integers  
        int[] numbers = {1, 2, 3, 4, 5};  
  
        // Access and print elements of the array  
        System.out.println("Element at index 0: " + numbers[0]);  
        System.out.println("Element at index 1: " + numbers[1]);  
        System.out.println("Element at index 2: " + numbers[2]);  
        System.out.println("Element at index 3: " + numbers[3]);  
        System.out.println("Element at index 4: " + numbers[4]);  
    }  
}
```

CONCEPT DIAGRAM



Displaying array elements using for loop

```
public class Arrays{  
    public static void main(String args[]){  
        int Arr[] = {26, 24, 32, 47, 75, 83, 91, 12};  
        for(int i=0; i<=4; i++){  
            System.out.println("The element at index "+i+" is "  
+Arr[i]);  
        }  
    }  
}  
  
-----  
-----  
  
// Output
```

The element at index 0 is 26
The element at index 1 is 24
The element at index 2 is 32
The element at index 3 is 47
The element at index 4 is 75

Here we have initiated the loop from 0 till index 4 that's why the loop printed 5 elements from the array.

Multidimensional Arrays

A multidimensional array is an array of arrays. This means that each element of the array is itself an array.

Java supports two-dimensional arrays, which are the most common type of multidimensional arrays. You can think of a two-dimensional array as a matrix or a table with rows and columns.

```
public class MultidimensionalArrayExample {
    public static void main(String[] args) {
        // Declare and initialize a 2D array (matrix) of integers
        int[][] matrix = {
            {1, 2, 3},
            {4, 5, 6},
            {7, 8, 9}
        };

        // Print the 2D array using nested loops
        System.out.println("2D Array (Matrix):");

        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[i].length; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println(); // Move to the next line after each row
        }
    }
}
```

// Output

2D Array (Matrix):
1 2 3
4 5 6
7 8 9

1. matrix is a 2D array of integers, initialized with values.
2. The outer loop (`for (int i = 0; i < matrix.length; i++)`) iterates over the rows of the matrix.
3. The inner loop (`for (int j = 0; j < matrix[i].length; j++)`) iterates over the elements in each row.
4. The elements are accessed using the indices `i` and `j`, and they are printed using `System.out.print()`.