Name: Ujjwal Kulkarni

Class: SY-10

Roll No: 2213621

## Assignment No. 1

**Assignment Title:** A java program to demonstrate the concept of operators.

**Aim**: Write a Java program that accepts four integers from the user and prints equal if all four are equal, and not equal otherwise.

**Pre-Requisites:** C/C++

**Objective**: The objective is to impart fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.

**Outcomes:** After learning this concept students will be able to,

1. Analyze the necessity for Object Oriented Programming paradigm over structured programming and become familiar with the fundamental concepts in OOP like encapsulation, Inheritance and Polymorphism
2. Design and develop java programs, analyse, and interpret object oriented data and report results.

## Theory:

**OPERATORS IN JAVA**

**Operator** in [Java](https://www.javatpoint.com/java-tutorial) is a symbol that is used to perform operations. For example: +, -, \*, / etc. There are many types of operators in Java which are given below:

* Unary Operator,
* Arithmetic Operator,
* Shift Operator,
* Relational Operator,
* Bitwise Operator,
* Logical Operator,
* Ternary Operator and
* Assignment Operator.

# Java Operator Precedence

|  |  |  |
| --- | --- | --- |
| **Operator Type** | **Category** | **Precedence** |
| Unary | postfix | *expr*++ *expr*-- |
| prefix | ++*expr* --*expr* +*expr* -*expr* ~ ! |
| Arithmetic | multiplicative | \* / % |
| additive | + - |
| Shift | shift | << >> >>> |
| Relational | comparison | < > <= >= instance of |
| equality | == != |
| Bitwise | bitwise AND | & |
| bitwise exclusive OR | ^ |
| bitwise inclusive OR | | |
| Logical | logical AND | && |
| logical OR | || |
| Ternary | ternary | ? : |
| Assignment | assignment | = += -= \*= /= %= &= ^= |= <<= >>= >>>= |

**Java Arithmetic Operators**

Java arithmetic operators are used to perform addition, subtraction, multiplication, and division. They act as basic mathematical operations.

Java User Input

The Scanner class is used to get user input, and it is found in the java.util package.

To use the Scanner class, create an object of the class and use any of the available methods found in the Scanner class documentation. In our example, we will use

the nextLine() method, which is used to read Strings

## Example

import java.util.Scanner; // Import the Scanner class

class Main {

public static void main(String[] args) {

Scanner myObj = new Scanner(System.in); // Create a Scanner object System.out.println("Enter username");

String userName = myObj.nextLine(); // Read user input System.out.println("Username is: " + userName); // Output user input

}

}

# INPUT TYPES

In the example above, we used the nextLine() method, which is used to read Strings. To read other types, look at the table below:

|  |  |
| --- | --- |
| **Method** | **Description** |
| nextBoolean() | Reads a boolean value from the user |
| nextByte() | Reads a byte value from the user |
| nextDouble() | Reads a double value from the user |
| nextFloat() | Reads a float value from the user |
| nextInt() | Reads a int value from the user |
| nextLine() | Reads a String value from the user |
| nextLong() | Reads a long value from the user |
| nextShort() | Reads a short value from the user |

## Java Arithmetic Operator Example

1. **public class** OperatorExample{
2. **public static void** main(String args[]){
3. **int** a=10;
4. **int** b=5;
5. System.out.println(a+b);//15
6. System.out.println(a-b);//5
7. System.out.println(a\*b);//50
8. System.out.println(a/b);//2
9. System.out.println(a%b);//0 10. }}

**Output:**

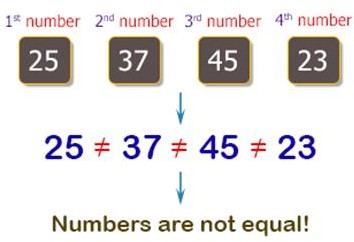
15

5

50

2

0



**Fig. Acceptance of four integers from user and checking there equality**

## Algorithm/Steps:

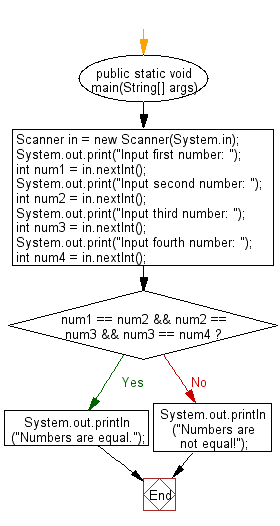
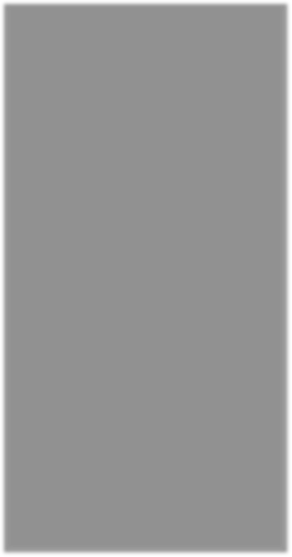
Step 1: Start

Step 2: Accept four integer numbers from user.

Step 3: Check the equality of numbers using assignment and AND operator. Step 4: If all numbers are equal Print output ”All four integers are equal” Step 5: Else Print output ”All four integers are not equal”

Step 6: Stop

## Flowchart:



Program Code:

import java.util.Scanner; public class Solution {

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

System.out.print("Input first number: "); int num1 = in.nextInt();

System.out.print("Input second number: "); int num2 = in.nextInt();

System.out.print("Input third number: "); int num3 = in.nextInt();

System.out.print("Input fourth number: "); int num4 = in.nextInt();

if (num1 == num2 && num2 == num3 && num3 == num4)

{

}

else

}

}

}

System.out.println("Numbers are equal.");

{

System.out.println("Numbers are not equal!");

Sample Output:

Input first number: 45 Input second number: 37 Input third number: 85 Input fourth number: 83 Numbers are not equal!

**Conclusion:** Thus we implemented a Java program that accepts four integers from the user and prints equal if all four are equal, and not equal otherwise.

## Frequently Asked Questions:

* 1. What is Class?

– A class can be defined as a template/blueprint that describes the behavior/state that the object of its type support.

* 1. What is object-oriented programming?
     + Object-oriented programming (OOP) is a programming paradigm that organizes code into objects that have properties (attributes) and methods (behaviors) that operate on those properties. In OOP, code is written in terms of objects and their interactions with each other, rather than in terms of actions to be taken on data.
  2. How to invoke methods in Java?
     + In Java, to invoke a method, you need to follow these steps:

1. Create an instance of the class that defines the method or use a static method if it's defined as a static method.
2. Use the dot operator (.) to access the method from the object or class name.
3. Pass any required arguments to the method within the parentheses.
4. If the method returns a value, you can assign it to a variable or use it directly in your code.
5. Write syntax for Scanner class in java.

– Syntax

import java.util.Scanner;

Scanner scannerObj = new Scanner(System.in); Object-oriented

1. Which operators are supported by Java? List and Brief it.

|  |  |  |
| --- | --- | --- |
| **Operator Type** | **Category** | **Precedence** |
| Unary | postfix | *expr*++ *expr*-- |
| prefix | ++*expr* --*expr* +*expr* -*expr* ~ ! |
| Arithmetic | multiplicative | \* / % |
| additive | + - |
| Shift | shift | << >> >>> |
| Relational | comparison | < > <= >= instance of |
| equality | == != |
| Bitwise | bitwise AND | & |
| bitwise exclusive OR | ^ |
| bitwise inclusive OR | | |
| Logical | logical AND | && |
| logical OR | || |
| Ternary | ternary | ? : |
| Assignment | assignment | = += -= \*= /= %= &= ^= |= <<= >>= >>>= |

1. What is Operator Precedence?

– Operator precedence refers to the order in which operators are evaluated in an expression that contains multiple operators. In programming languages, operators have different levels of precedence, which determines the order in which they are executed,