

# Introduction to Java



Mahmudul Hasan

QA Automation Engineer

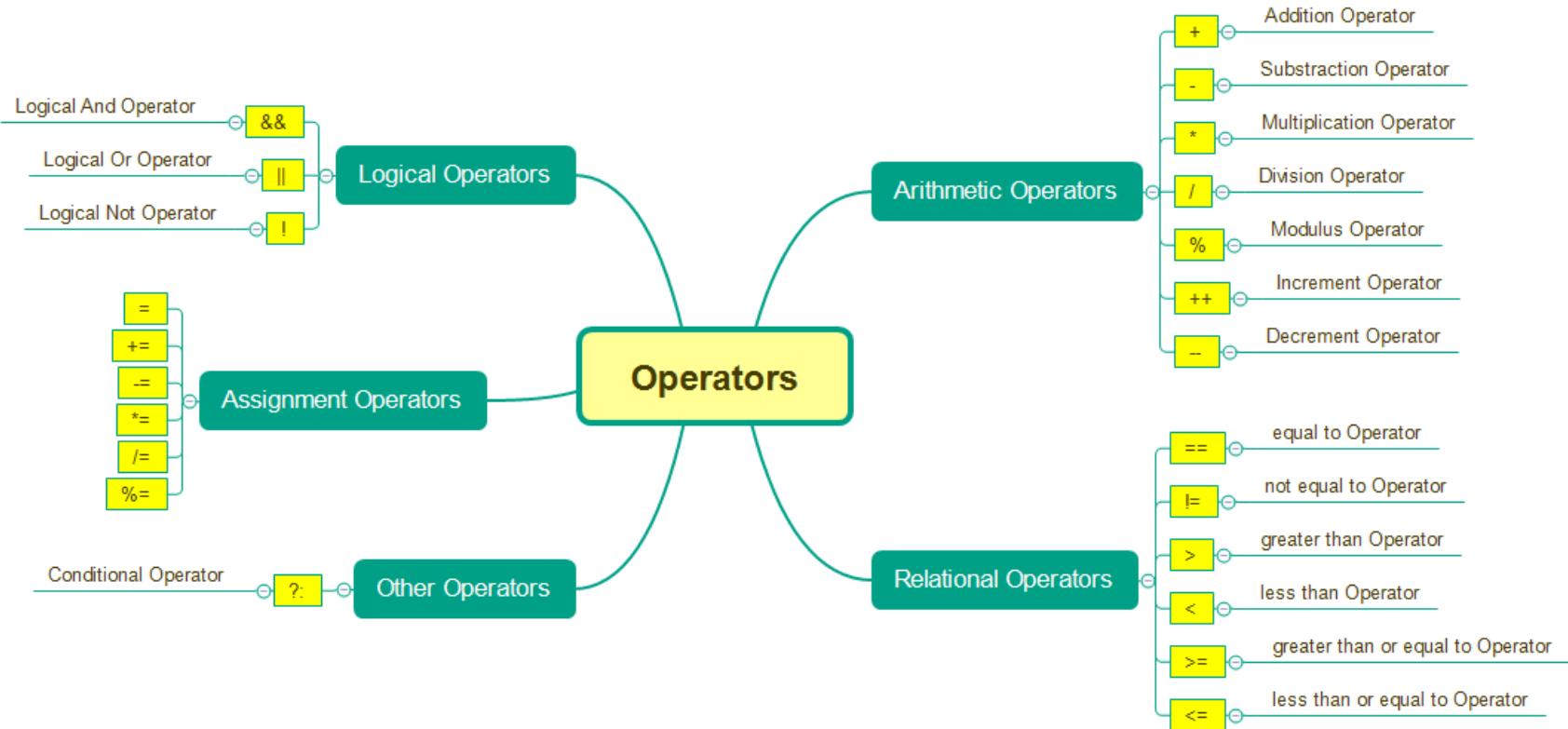
Email: [mahmudul006@gmail.com](mailto:mahmudul006@gmail.com)

# Outline

- Operators
- Types of Operators
- Arithmetic Operators
- Assignment Operators
- Relational/Comparison Operators
- Logical Operators
- Java Control Statements
- IF/ELSE Statements
- IF/ELSE Statements Code Example
- ELSE IF Statements Code Example
- Switch Statements
- Switch Statements Code Example

# Operators

Operators are symbols that perform operations on variables and values.



# Types of Operators

- Arithmetic Operators
- Assignment Operators
- Relational Operators
- Logical Operators
- Unary Operators
- Ternary Operators
- Shift Operators
- Bitwise Operators

Article 1: [Operators in Java](#)

Article 2: [Operators in Java](#)

# Arithmetic Operators

Arithmetic operators are used to perform common mathematical operations.

Operator	Description	Example
+ Addition	Adds values on either side of the operator	$A+B=30$
- Subtraction	Subtracts the right-hand operator with left-hand operator	$A-B=-10$
* Multiplication	Multiplies values on either side of the operator	$A*B=200$
/ Division	Divides left hand operand with right hand operator	$A/B=0$
% Modulus	Divides left hand operand by right hand operand and returns remainder	$A\%B=0$

# Assignment Operators

Assignment operators are used to assign values to variables.

Operator	Description	Example
=	Assigns values from right side operands to left side operand	<code>c = a + b</code>
+=	It adds right operand to the left operand and assigns the result to left operand	<code>c += a</code>
-=	It subtracts right operand from the left operand and assigns the result to left operand	<code>c -= a</code>
*=	It multiplies right operand with the left operand and assigns the result to left operand	<code>c *= a</code>
/=	It divides left operand with the right operand and assigns the result to left operand	<code>c /= a</code>
%=	It takes modulus using two operands and assigns the result to left operand	<code>c %= a</code>
^=	Performs exponential (power) calculation on operators and assign value to the left operand	<code>c ^= a</code>

# Relational/Comparison Operators

Relational operators are used to compare two values.

Operator	Description	Example
<code>==</code>	If the values of two operands are equal, then the condition becomes true.	$(A == B)$ is not true
<code>!=</code>	If the values of two operands are not equal, then condition becomes true.	$(A != B)$ is true
<code>&gt;</code>	If the value of the left operand is greater than the value of right operand, then condition becomes true.	$(a > b)$ is not true
<code>&lt;</code>	If the value of the left operand is less than the value of right operand, then condition becomes true.	$(a < b)$ is true
<code>&gt;=</code>	If the value of the left operand is greater than or equal to the value of the right operand, then condition becomes true.	$(a >= b)$ is not true
<code>&lt;=</code>	If the value of the left operand is less than or equal to the value of right operand, then condition becomes true.	$(a <= b)$ is true

# Logical Operators

Logical operators are used to check whether an expression is true or false .

Operator	Description	Example
<code>&amp;&amp; (and)</code>	True if both the operands is true	<code>a&lt;10 &amp;&amp; a&lt;20</code>
<code>   (or)</code>	True if either of the operands is true	<code>a&lt;10    a&lt;20</code>
<code>! (not)</code>	True if an operand is false (complements the operand)	<code>!(x&lt;10 &amp;&amp; a&lt;20)</code>

# Java Control Statements

In programming, control flow is the order in which code is executed

- **Decision Making statements**

- *if statements*
- *switch statement*

- **Loop statements**

- *do while loop*
- *while loop*
- *for loop*
- *for-each loop*

- **Jump statements**

- *break statement*
- *continue statement*

Article 1: Java Control Statements

# IF/ELSE Statements

if statement is used to test the condition. It checks boolean condition: true or false. There are various types of if statement in Java.

- if statement
- if-else statement
- If-else-if ladder
- nested if statement

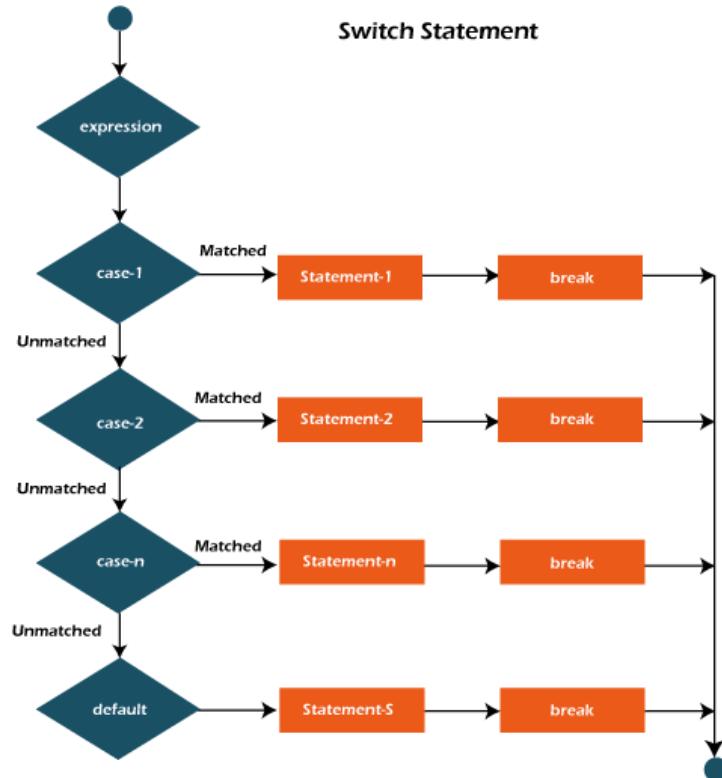
# IF/ELSE Statements Code Example

```
public class Main {  
  
    public static void main(String[] args) {  
        boolean isLightOn = false;  
  
        if(isLightOn) {  
            System.out.println("The light is turned on");  
        } else {  
            System.out.println("The light is turned off");  
        }  
    }  
}
```

# ELSE IF Statements Code Example

```
public class Main {  
  
    public static void main(String[] args) {  
        int number = 50;  
  
        if(number <= 20) {  
            System.out.println("Number is less than or equal to 20");  
        } else if(number < 40) {  
            System.out.println("Number is between 20 and 40");  
        } else {  
            System.out.println("Number is greater than or equal 40");  
        }  
    }  
}
```

# Switch Statements



# Switch Statements Code Example

```
int day = 4;
switch (day) {
    case 1:
        System.out.println("Monday");
        break;
    case 2:
        System.out.println("Tuesday");
        break;
    case 3:
        System.out.println("Wednesday");
        break;
    case 7:
        System.out.println("Sunday");
        break;
    default:
        System.out.println("Invalid day");
        break;
}
```

Thank you ❤

⌚ qa-june-2024-automation-with-java-slides