## Practical No.: 1

**Programs on Operators, Arithmetic Promotion, Method Calling.**

**Goal:** Performing Basic operations on operators check for arithmetic promotion .

**Prerequisite:** Algorithm, Basic of Java

**Software Requirements:** Netbean, JDK 1.8.

# Theory / Analysis:

# Operators in Java:

# Java provides many types of operators which can be used according to the need. They are classified based on the functionality they provide. Some of the types are-

1. Arithmetic operators
2. Assignment operators
3. Comparison operators
4. Logical operators
5. Bitwise operators

**Arithmetic Operators:** They are used to perform simple arithmetic operations on primitive data types.

* **\* :** Multiplication
* **/ :** Division
* **% :** Modulo
* **+ :** Addition
* **– :** Subtraction

**Unary Operators:** Unary operators needs only one operand. They are used to increment, decrement or negate a value.

* **– :Unary minus**, used for negating the values.
* **+ :Unary plus**, used for giving positive values. Only used when deliberately converting a negative value to positive.
* **++ :Increment operator**, used for incrementing the value by 1. There are two varieties of increment operator.
  + **Post-Increment :** Value is first used for computing the result and then incremented.
  + **Pre-Increment :** Value is incremented first and then result is computed.
* **- - : Decrement operator**, used for decrementing the value by 1. There are two varieties of decrement operator.
  + **Post-decrement :** Value is first used for computing the result and then decremented.
  + **Pre-Decrement :** Value is decremented first and then result is computed.
* **! : Logical not operator**, used for inverting a boolean value.

**Assignment Operator : ‘=’** Assignment operator is used to assign a value to any variable. It has a right to left associativity, i.e value given on right hand side of operator is assigned to the variable on the left and therefore right hand side value must be declared before using it or should be a constant.  
General format of assignment operator is,

variable **=** value;

In many cases assignment operator can be combined with other operators to build a shorter version of statement called **Compound Statement**. For example, instead of a **=** a+5 , we can write a **+=** 5.

* **+=**, for adding left operand with right operand and then assigning it to variable on the left.
* **-=**, for subtracting left operand with right operand and then assigning it to variable on the left.
* **\*=**, for multiplying left operand with right operand and then assigning it to variable on the left.
* **/=**, for dividing left operand with right operand and then assigning it to variable on the left.
* **%=**, for assigning modulo of left operand with right operand and then assigning it to variable on the left.

**Relational Operators :** These operators are used to check for relations like equality, greater than, less than. They return boolean result after the comparison and are extensively used in looping statements as well as conditional if else statements. General format is,

variable **relation\_operator** value

Some of the relational operators are-

* **== , Equal to :** returns true of left hand side is equal to right hand side.
* **!= , Not Equal to :** returns true of left hand side is not equal to right hand side.
* **< , less than :** returns true of left hand side is less than right hand side.
* **<= , less than or equal to :** returns true of left hand side is less than or equal to right hand side.
* **> , Greater than :** returns true of left hand side is greater than right hand side.
* **>= , Greater than or equal to:** returns true of left hand side is greater than or equal to right hand side.

**Logical Operators :** These operators are used to perform “logical AND” and “logical OR” operation, i.e. the function similar to AND gate and OR gate in digital electronics. One thing to keep in mind is the second condition is not evaluated if the first one is false, i.e. it has short-circuiting effect. Used extensively to test for several conditions for making a decision.

**Conditional operators are-**

* **&& , Logical AND :** returns true when both conditions are true.
* **|| , Logical OR :** returns true if at least one condition is true.
* **Ternary operator :** Ternary operator is a shorthand version of if-else statement. It has three operands and hence the name ternary. General format is-
* condition **?** if true **:** if false
* The above statement means that if the condition evaluates to true, then execute the statements after the ‘?’ else execute the statements after the ‘:’.

**Bitwise Operators :** These operators are used to perform manipulation of individual bits of a number. They can be used with any of the integer types. They are used when performing update and query operations of Binary indexed tree.

* **& , Bitwise AND operator:** returns bit by bit AND of input values.
* **| , Bitwise OR operator:** returns bit by bit OR of input values.
* **^ , Bitwise XOR operator:** returns bit by bit XOR of input values.
* **~ , Bitwise Complement Operator:** This is a unary operator which returns the one’s compliment representation of the input value, i.e. with all bits inversed.

**What is Arithmetic Promotion in Java?**   
The arithmetic promotion in Java happens when you apply an arithmetic operation on two variables with different data-types. In this case the compiler will convert the data type of one operand in the binary arithmetic operation to the type of the other operand.   
  
**Here are some rules about arithmetic promotion in Java:**

* If one operand is of type "double", then the other operand is promoted to the type double.
* If Both operands are of type less than "int" , for example (byte,short,char...) .Both operands are promoted to type integer.
* If one of operands is type float and the other is not double .The other operand is converted to type float.
* If one of operands is type long and the other is not double or float .The other operand is converted to type long.
* If no operand of type (double,float ,long ) , then convert to type integer .

**Implementation:**

Source code of the program is in Java.

**Conclusion:**