Unary Operators

Unary Operators

Unary Operators in Java are the types of operators that require only one operand. They form various operations on single operands such as incrementing or decrementing the value by one, negation of an expression, or inverting the value of a boolean. Let's understand the various unary operators with an example.

(i) Unary minus operator (-): This operator can be used to convert a negative value into a positive value or positive value into a negative value.

Example:

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

      // Convert a positive value
      // into a negative value
      int num1 = 10;
      num1 = -num1;
      System.out.println("Negative Value : " +num1);


      // Convert a negative value
      // into a positive value
      int num2 = -20;
      num2 = -num2;
      System.out.println("Positive Value : " +num2);
  }
}
```

Output:

```
Negative Value : -10
Positive Value : 20
```

(ii) Unary NOT Operator (!): This operator is used to convert the true to false and vice versa.

Example:

```
public class UnaryOperator {
   public static void main(String args[]) {
      int a = 10, b = 20;

       // Without using NOT unary operator.
      System.out.println("(a < b) = " + (a < b));

      // Using unary NOT operator.
      System.out.println("!(a < b) = " + !(a < b));
  }
}
```

Output:

```
(a < b) = true
!(a < b) = false
```

(iii) Increment Operator (++): This operator is used to increment the value by 1. There are two types of increment operator

1. **Post-increment operator: Post increment operator is used to increment the value of the variable after it has been evaluated for use in the expression.**
2. **Pre-increment operator: Pre increment operator is used to increment the value of the variable before it's evaluated in the expression.**

Example:

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


      // Initialize the variable
      int num = 10;


      // first print 10, then number is
      // increment to 11
      System.out.println("Post increment = " + num++);


      // num was 11, incremented to 12 and print
      System.out.println("Pre increment = " + ++num);
  }
}
```

Output:

```
Post increment = 10
Pre increment = 12
```

(iv) Decrement Operator (--): This operator is used to decrement the value by 1. There are two types of decrement operators.

1. **Post-decrement operator: Post decrement operator is used to decrement the value of the variable after it has been evaluated for use in the expression.**
2. **Pre-decrement operator: Pre decrement operator is used to decrement the value of the variable before it's evaluated in the expression.**

Example:

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


      // Initialize the variable
      int num = 10;


      // first print 10, then number is
      // decrement to 9
      System.out.println("Post decrement = " + num--);


      // num was 9, decremented to 8 and print
      System.out.println("Pre increment = " + --num);
  }
}
```

Output:

```
Post decrement = 10
Pre decrement = 8
```

v) Bitwise Complement (~): This operator is used to return the one's complement representations of the input value.

Example:

```
public class BitwiseComplement {
   public static void main(String args[]) {
      int num1 = 7;
      int num2 = -8;
      // Performing bitwise complement
      System.out.println(num1 +" 's bitwise complement = " + ~num1);
      System.out.println(num2 +" 's bitwise complement = " + ~num2);
  }
}
```

Output:

```
7's bitwise complement = -8
-8's bitwise complement = 7
```