Unary Operators

Unary operators are those operators in Java that only need a single operand to perform any function. They work on the same principal as unary operations in mathematics.

Increment Unary Operator

It increments the value by 1 where ++x = x+1.

Decrement Unary Operator

It decrements the value by 1 where --x = x-1.

5. Logical Complement

It logically inverts the value of a boolean like if x = true, then !x will be false.

Increment Operator (++)

The increment (++) operator (also known as increment unary operator) in Java is used to increase the value of a variable by 1. Since it is a type of a unary operator, it can be used with a single operand.

Syntax

The syntax for increment operator is a pair of addition signs ie;

```
++x;
x++;
```

The operator can be applied either before or after the variable. Both will have the same increment of 1. However, they both have separate uses and can be categorized as the following types.

- Pre-Increment Operator
- Post-Increment Operator

Example

```
public class IncrementOperator {

     public static void main(String[] args) {

int variable = 15;
System.out.println("Original value of variable = " + variable);

// after using increment operator
variable++;    //  increments 1, variable = 16
System.out.println("variable++ = " + variable);

          ++variable;       //  increments 1, variable 17
          System.out.println("++variable = " +
variable);
     }
}
```

Output

```
Original value of the variable = 15
variable++ = 16
++variable = 17
```

Pre-Increment Operator (++x;)

If the increment operator (++) is specified before the variable like a prefix (++x), then it is called pre-increment operator. In this case, the value of the

variable is first incremented by 1, and then further computations are performed.

Example

```
public class PreIncrementOperator {

    public static void main(String[] args) {

        int variable = 5;
        System.out.println("Original value of the
variable = " + variable);

        // using pre-increment operator
        int preIncrement = ++variable;

        System.out.println("variable = " + variable);
        System.out.println("preIncrement = " +
preIncrement);
        System.out.println("++preIncrement = " +
++preIncrement);
    }
}
```

Output

```
Original value of the variable = 5
variable = 6
preIncrement = 6
++preIncrement = 7
```

Post-Increment Operator (x++;)

If the increment operator (++) is specified after the variable like a postfix (x++), then it is called post-increment operator. In this case, the original

value of the variable (without increment) is used for computations and then it is incremented by 1.

Example

```
public class PostIncrementOperator {

    public static void main(String[] args) {

        int variable = 100;
        System.out.println("Original value of the
variable = " + variable);

        // using post-increment operator
        int postIncrement = variable++; //
postIncrement = 100, variable = 101

        System.out.println("postIncrement = " +
postIncrement);
        System.out.println("variable = " + variable +
"\n");

            // postIncrement = 101
        System.out.println("postIncrement++ = " +
postIncrement++);
            // postIncrement = 102
        System.out.println("postIncrement++ = " +
postIncrement++);
            // postIncrement = 103
        System.out.println("postIncrement++ = " +
postIncrement++);

        System.out.println("\npostIncrement = " +
postIncrement);
    }
}
```

Output

```
Original variable = 100
postIncrement = 100
variable = 101

postIncrement++ = 100
postIncrement++ = 101
postIncrement++ = 102

postIncrement = 103
```

Decrement Operator (--)

Decrement as the name implies is used to reduce the value of a variable by 1. It is also one of the unary operator types, so it can be used with a single operand.

Syntax

The syntax for decrement operator is a pair of negative signs ie;

```
--x;
x--;
```

Just like the increment operator, the decrement (--) operator can also be applied before and after the variable. Both will result in the same decrement of 1. They both have distinct uses and can be diverged in the further types.

- Pre-Decrement Operator
- Post-Decrement Operator

Pre-Decrement Operator (--x;)

If the decrement operator (--) is mentioned before the variable like a prefix (--x), then it is called a pre-decrement operator. For this case, the value of the variable is first decremented by 1, and then other computations are performed.

Example

```
public class PreDecrementOperator {

     public static void main(String[] args) {

          int variable = 11;
          System.out.println("Original value of the
variable = " + variable);

          // using preDecrement operator
          int preDecrement = --variable;

             // variable = 10
          System.out.println("variable = " + variable);
             // preDecrement = 10
          System.out.println("preDecrement = " +
preDecrement);
             // preDecrement = 9
          System.out.println("--preDecrement = " +
--preDecrement);       }
}
```

Output

```
Original value of the variable = 11
variable = 10
preDecrement = 10
--preDecrement = 9
```

Post-Decrement Operator (x--;)

If the decrement operator (--) is mentioned after the variable like a postfix (x--), then it is called a post-decrement operator. For this case, the original value of the variable (without decrement) is used for computations and then it is decremented by 1.

Example

```
public class PostDecrementOperator {

    public static void main(String[] args) {

        int variable = 75;
        System.out.println("Original value of the
variable = " + variable);

        // using postDecrement operator
           // postDecrement = 75, variable = 74
        int postDecrement = variable--;
        System.out.println("postDecrement = " +
postDecrement);
        System.out.println("variable = " + variable +
"\n");
        // postDecrement = 74
        System.out.println("postDecrement-- = " +
postDecrement--);
           // postDecrement = 73
        System.out.println("postDecrement-- = " +
postDecrement--);
           // postDecrement = 72
        System.out.println("postDecrement-- = " +
postDecrement--);

        System.out.println("\npostDecrement = " +
postDecrement);
    }
}
```

```
Original value of the variable = 75
postDecrement = 75
variable = 74

postDecrement-- = 75
postDecrement-- = 74
postDecrement-- = 73

postDecrement = 72
```

.