# OPERATORS

ChandraSekhar(CS) Baratam

# ARITHMETIC OPERATOR

| Operator | Meaning | Example |
|---|---|---|
| + | Addition | 8 + 10 |
| - | Subtraction | 10 – 8 |
| * | Multiplication | 12 * 2 |
| / | Division | 10 / 5 |
| % | Modulus | 10 % 6 |

# SHORTCUT ASSIGNMENT OPERATOR

**Shortcut Assignment Operators**

| Operator | Use | Description |
|---|---|---|
| += | op1 += op2 | Equivalent to op1 = op1 + op2 |
| -= | op1 -= op2 | Equivalent to op1 = op1 - op2 |
| *= | op1 *= op2 | Equivalent to op1 = op1 * op2 |
| /= | op1 /= op2 | Equivalent to op1 = op1 / op2 |
| %= | op1 %= op2 | Equivalent to op1 = op1 % op2 |
| &= | op1 &= op2 | Equivalent to op1 = op1 & op2 |
| \|= | op1 \|= op2 | Equivalent to op1 = op1 \| op2 |
| ^= | op1 ^= op2 | Equivalent to op1 = op1 ^ op2 |
| <<= | op1 <<= op2 | Equivalent to op1 = op1 << op2 |
| >>= | op1 >>= op2 | Equivalent to op1 = op1 >> op2 |
| >>>= | op1 >>>= op2 | Equivalent to op1 = op1 >>> op2 |

# RELATIONAL OPERATOR

| Operator | Use | Description |
|---|---|---|
| > | op1 > op2 | op1 is greater than op2 |
| >= | op1 >= op2 | op1 is greater than or equal to op2 |
| < | op1 < op2 | op1 is less than op2 |
| <= | op1 <= op2 | op1 is less than or equal to op2 |
| == | op1 == op2 | op1 and op2 are equal |
| != | op1 != op2 | op1 and op2 are not equal |

# LOGICAL OPERATOR

| Operator | Name | Example | Description |
|---|---|---|---|
| && | AND operator | A && B; | If either of the values are zero the value of the expression is zero, otherwise the value of the expression is 1. If the left hand value is zero, then the right hand value is not considered. |
| \|\| | OR operator | A \|\| B; | If both of the values are zero then the value of the expression is 0 otherwise the value of the expression is 1. If the left hand value is non-zero, then the right hand value is not considered. |
| ! | NOT operator | !A; | Not operator is applied to a non-zero value then the value is zero, if it is applied to a zero value, the value is 1 |

# INCREMENT OR DECREMENT OPERATOR

| Operator | Purpose | Example | Notes |
|---|---|---|---|
| ++ | Pre-increment (++variable) | int i = 6;<br>int j = ++i;<br>i is 7, j is 7 | |
| | Post-increment (++variable) | int i = 6;<br>int j = i++;<br>i is 7, j is 6 | The value of i is assigned to j before i is incremented. Therefore, j is assigned 6. |
| -- | Pre-decrement (--variable) | int i = 6;<br>int j = --i;<br>i is 5, j is 5 | |
| | Post-decrement (--variable) | int i = 6;<br>int j = i--;<br>i is 5, j is 6 | The value of i is assigned to j before i is decremented. Therefore, j is assigned 6. |

# TERNARY OPERATOR

result = testCondition ? value1 : value2