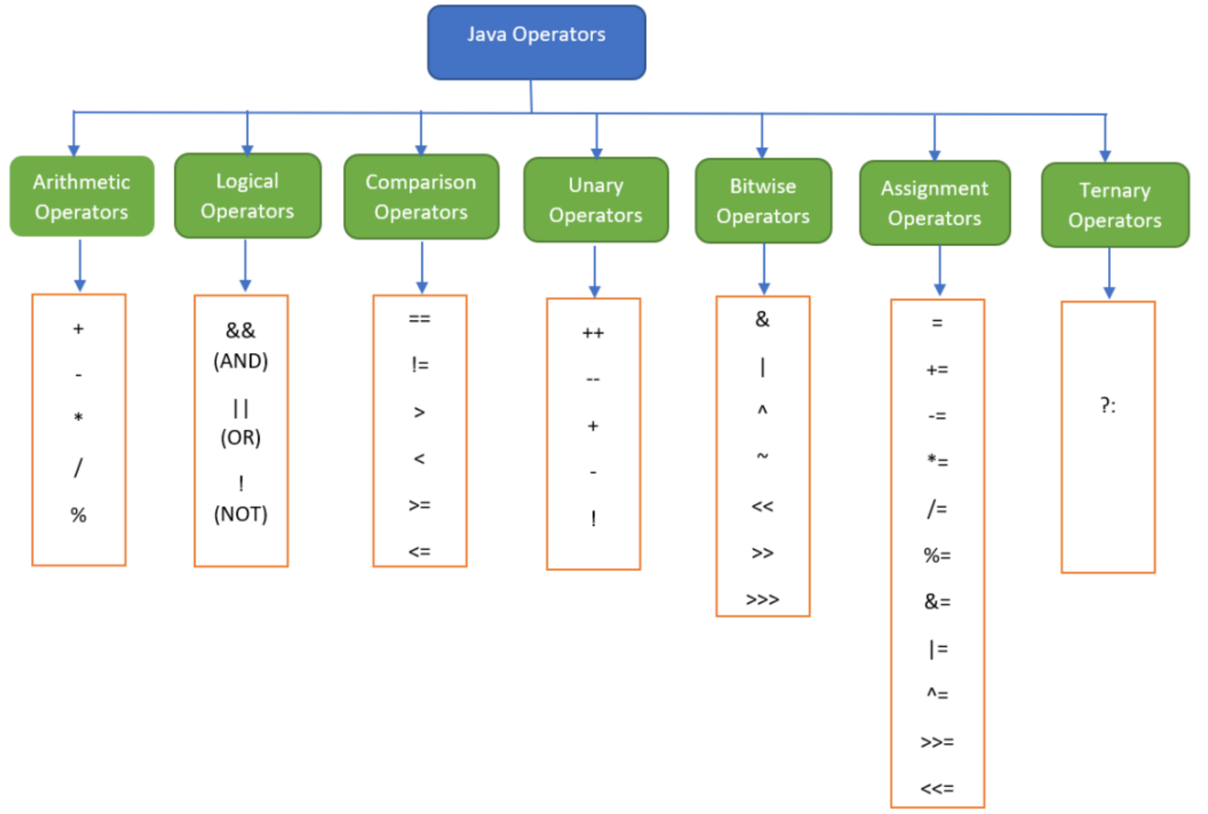

```
// Arithmetic operators

int a = 10;

int b = 5;

int sum = a + b; // sum is now 15

int difference = a - b; // difference is now 5

int product = a * b; // product is now 50

int quotient = a / b; // quotient is now 2

int remainder = a % b; // remainder is now 0


// Assignment operators

int c = 10;

c += 5; // c is now 15

c -= 5; // c is now 10

c *= 5; // c is now 50

c /= 5; // c is now 10

c %= 5; // c is now 0
```

```
// Relational operators
int d = 10;
int e = 5;
boolean isEqualTo = d == e; // isEqualTo is now false
boolean isNotEqualTo = d != e; // isNotEqualTo is now true
boolean isGreaterThan = d > e; // isGreaterThan is now true
boolean isLessThan = d < e; // isLessThan is now false
boolean isGreaterThanOrEqualTo = d >= e; // isGreaterThanOrEqualTo is now true
boolean isLessThanOrEqualTo = d <= e; // isLessThanOrEqualTo is now false

// Logical operators
boolean isTrue = true;
boolean isFalse = false;
boolean andResult = isTrue && isFalse; // andResult is now false
boolean orResult = isTrue || isFalse; // orResult is now true
boolean notResult = !isTrue; // notResult is now false

// Bitwise operators
int f = 10;
int g = 5;
int andResult = f & g; // andResult is now 0
int orResult = f | g; // orResult is now 15
int exclusiveOrResult = f ^ g; // exclusiveOrResult is now 15
int leftShiftResult = f << 1; // leftShiftResult is now 20
int rightShiftResult = f >> 1; // rightShiftResult is now 5

// Unary operators
int h = 10;
int incrementResult = ++h; // incrementResult is now 11
int decrementResult = --h; // decrementResult is now 9
int negationResult = -h; // negationResult is now -9

// Ternary operator
int i = 10;
int j = 5;
int max = i > j ? i : j; // max is now 10

// Instanceof operator
Object o = new Object();
boolean isInstanceOfObject = o instanceof Object; // isInstanceOfObject is now true
```