



# ARITHMETIC OPERATORS

## C++ BASICS

---

*prepared by:*

**Gyro A. Madrona**

Electronics Engineer

# TOPIC OUTLINE

Addition

Subtraction

Multiplication

Division

Modulo

Increment

Decrement



# ARITHMETIC OPERATORS



# ARITHMETIC OPERATORS

---

Arithmetic operators are used to perform basic mathematical operations on numeric values.

These operators are fundamental to performing calculations and manipulating data in programs.

Unary Operators operates with a single operand (e.g., ++, --).

Binary Operators operates with two operands (e.g., +, -, \*, %).



# ADDITION

---

Example:

```
int a = 5;
```

```
int b = 3;
```

```
int result = a + b;
```

```
// result = 8
```

The addition (+) operator **adds** two operands.



# SUBTRACTION

---

Example:

```
int a = 10;
```

```
int b = 4;
```

```
int result = a - b;
```

```
// result = 6
```

The subtraction (-) operator **subtracts** the second operand from the first.



# MULTIPLICATION

---

The multiplication (\*) operator multiplies two operands.

Example:

```
int a = 7;
```

```
int b = 6;
```

```
int result = a * b;
```

```
// result = 42
```



# DIVISION

---

The division // operator **divides** the first operand by the second.

Example:

```
int a = 10;
```

```
int b = 3;
```

```
float result = a / b;
```

```
// result = 3.333
```





# MODULO

---

The modulo (%) operator returns the **remainder** of the division of the first operand by the second.

Example:

```
int a = 10;
```

```
int b = 3;
```

```
float result = a % b;
```

```
// result = 1
```



# INCREMENT

---

Example:

```
int x = 5;
```

```
x++;
```

```
// x = 6
```

The increment (++) operator increases the value of a variable by 1.



# DECREMENT

---

Example:

```
int x = 5;
```

```
x--;
```

```
// x = 4
```

The decrement (--) operator decreases the value of a variable by 1.



## EXERCISE

---

Determine the output of this code snippet:

```
int a = 2;  
  
int b = 4;  
  
int c = 0;  
  
c = (a + b) / a;  
  
cout << c;
```

output:

Determine the output of this code snippet:

```
int a = 2;  
  
int b = 4;  
  
int c = 0;  
  
c = (a + b) % a;  
  
cout << c;
```

output:



## EXERCISE

---

Determine the output of this code snippet:

```
int x = 5;  
  
int y = 3;  
  
int z = 0;  
  
z = (x * y) - (x + y);  
  
cout << z;
```

output:

Determine the output of this code snippet:

```
int p = 10;  
  
int q = 2;  
  
int r = 0;  
  
r = (p % q) + (p / q);  
  
cout << r;
```

output:



## EXERCISE

---

Determine the output of this code snippet:

```
char a = '2';
```

```
char b = '3';
```

```
char c = '4';
```

```
c = a + b;
```

```
cout << c;
```

output:

101

Output Explanation:

When you add two **char** variables, they are implicitly converted to their ASCII integer values before the addition.

ASCII of '2' is 50.

ASCII of '3' is 51.



## EXERCISE

---

Determine the output of this code snippet:

```
string a = "2";
```

```
string b = "3";
```

```
string c = "4";
```

```
c = a + b;
```

```
cout << c;
```

output:

23

Output Explanation:

The `+` operator performs string concatenation. `a + b` concatenates the string `"2"` and `"3"`, resulting in the string `"23"`.



# EXERCISE

---

Determine the output of this code snippet:

```
int a = 5;  
  
int b = a++;  
  
int c = ++a;  
  
cout << b << " " << c;
```

output:

5 7

Output Explanation:

**++a** is pre-increment: increment first then use.

**a++** is post-increment: use first then increment.





## EXERCISE

---

Determine the output of this code snippet:

```
int x = 10;  
  
int y = --x;  
  
int z = x--;  
  
cout << y << " " << z;
```

output:

Determine the output of this code snippet:

```
int x = 10;  
  
int y = --x;  
  
int z = ++x;  
  
cout << y << " " << z;
```

output:



## EXERCISE

---

Determine the output of this code snippet:

```
int p = 3;  
int q = p++ + ++p;  
cout << q;
```

output:

8

Output Explanation:

**p++** evaluates to **3** (current value of **p**), then increments **p** to **4**.

**++p** increments **p** to **5** and evaluates to **5**.

**q = 3 + 5.**



## EXERCISE

---

Determine the output of this code snippet:

```
int m = 6;  
  
int n = 2;  
  
int o = m-- - --n;  
  
cout << o
```

output:

Determine the output of this code snippet:

```
int d = 4;  
  
int e = d++ * --d;  
  
cout << e;
```

output:



# LABORATORY

