

C++ Stage 2 – Operators & Control Flow

1. Operators in C++

Operators are symbols that perform actions on values or variables.

A. Arithmetic Operators

Performs basic calculations.

Operator	Meaning	Example	Result
+	Addition	5 + 3	8
-	Subtraction	5 - 3	2
*	Multiplication	5 * 3	15
/	Division	10 / 2	5
%	Modulus	10 % 3	1

```
int a = 10, b = 3;  
cout << a + b;  
cout << a % b;
```

B. Assignment Operators

Used to store or update values in variables.

Operator	Meaning	Example
=	Assign	x = 5
+=	Add & assign	x += 3
-=	Subtract & assign	x -= 2
*=	Multiply & assign	x *= 2
/=	Divide & assign	x /= 2
%=	Modulus & assign	x %= 2

C. Comparison Operators

Used to compare two values.

Operator	Meaning	Example	Result
==	Equal	5 == 5	true
!=	Not equal	5 != 3	true
>	Greater than	5 > 3	true
<	Less than	3 < 5	true
>=	Greater or equal	5 >= 5	true
<=	Less or equal	3 <= 5	true

D. Logical Operators

Combine multiple conditions.

Operator	Meaning	Example	Result
&&	AND	(5>3 && 2<4)	true
			OR
!	NOT	!(5>3)	false

2. Decision Making (If Statements)

Decision-making statements allow the program to perform different instructions based on conditions.

A. If Statement

Executes instructions if a condition is true.

```
int num = 10;
if(num > 5) {
    cout << "Number is greater than 5";
}
```

B. If-Else Statement

Chooses between two options based on condition.

```
int num = 4;
if(num % 2 == 0) {
    cout << "Even";
} else {
```

```
    cout << "Odd";  
}
```

C. Nested If

An if statement inside another if statement to check multiple conditions.

```
int marks = 85;  
if(marks >= 50) {  
    if(marks >= 80) {  
        cout << "Grade A";  
    } else {  
        cout << "Passed";  
    }  
} else {  
    cout << "Failed";  
}
```

D. Switch Statement

Selects one option from multiple cases.

```
int day = 3;  
switch(day) {  
    case 1: cout << "Monday"; break;  
    case 2: cout << "Tuesday"; break;  
    case 3: cout << "Wednesday"; break;  
    default: cout << "Invalid day";  
}
```

3. Loops

Loops repeat instructions multiple times.

A. For Loop

Repeats a task a fixed number of times.

```
for(int i = 1; i <= 5; i++) {  
    cout << i << endl;  
}
```

B. While Loop

Repeats a task while a condition is true.

```
int i = 1;
while(i <= 5) {
    cout << i << endl;
    i++;
}
```

C. Do-While Loop

Runs instructions at least once, then repeats while condition is true.

```
int i = 1;
do {
    cout << i << endl;
    i++;
} while(i <= 5);
```

D. Break & Continue

- `break` → stops the loop immediately
- `continue` → skips current iteration and continues

```
for(int i=1; i<=5; i++) {
    if(i == 3) break;
    cout << i << endl;
}

for(int i=1; i<=5; i++) {
    if(i == 3) continue;
    cout << i << endl;
}
```