



Object-Oriented Programming in C++

PRESENTED BY – ANJALI SONA
DEPARTMENT OF COMPUTER SCIENCE



CONTAIN

- INTRODUCTION

- OPERATORS IN C++

1. Unary Operators'
2. Binary Operators
3. Ternary Operators

- TYPES OF OPERATORS

1. Arithmetic operator
2. Logical operators
3. Comparison operator
4. Assignment operators
5. Bitwise operators
6. Scope resolution operator(::)
7. Line feed operator
8. Field width operator



INTERODUCTION

Operators are the foundation of any programming language. We can define operators as symbols that help us to perform specific mathematical and logical computations on operands. In other words, we can say that an operator operates the operands.

For example, '+' is an operator used for addition, as shown below:
`c = a + b;`

Here, '+' is the operator known as the addition operator and 'a' and 'b' are operands. The addition operator tells the compiler to add both of the operands 'a' and 'b'. The functionality of the C/C++ programming language is incomplete without the use of operators.



Operators in C++

- ▶ The symbols that are used in C++ programs to form an expression are known as operators. C++ has a rich set of operators including all language's operators and also some new operators. There are three categories, of operators in C++, These are
 1. Unary Operators'
 2. Binary Operators
 3. Ternary Operators

Operators in C

	Operator	Type
Unary operator →	+, -, ++, --	Unary operator
Binary operator {	+, -, *, /, %	Arithmetic operator
	<, <=, >, >=, ==, !=	Relational operator
	&&, , !	Logical operator
	&, , <<, >>, ~, ^	Bitwise operator
	=, +=, -=, *=, /=, %=	Assignment operator
Ternary operator →	?:	Ternary or conditional operator



Unary Operators

- ▶ The operators that operate a single operand to form an expression are known as unary operator. The operators like ++ (increment) operator, -- (decrement) operator etc. are the part of unary operators.



Binary Operators

- The operators that operate two or more operands are known as binary operators. The operators like + , - , * , / , % etc are binary operators.
- Ex-

$a+b$, $a-b$, $a*b$, a/b etc



Ternary Operators

- ▶ The operator that operates minimum or maximum three operands is known as ternary operator. there is only one ternary operator available in C++. The operator `?:` is the only available ternary operator i.e used as a substitute of if-else statement.

Ex

`a>b?a:b`



Types of Operators in C++

- Arithmetic operator
- Logical operators
- Comparison operator
- Assignment operators
- Bitwise operators
- Scope resolution operator(::)



Arithmetic Operators

- ▶ The operators that help the programmer in mathematical calculations are known as arithmetic operators. Arithmetic operators include (+) for addition, (-) for subtraction, (/) for division, (*) for multiplication etc.
- ▶ Ex

$$2+5=7$$



Logical Operators

- ▶ The operators that help the programmer to connect (combine) two or more expressions, are known as logical operators.
- ▶ Logical operators include:
 1. `&&` logical AND
 2. `||` logical OR
 3. `!` logical NOT



Comparison Operators

- ▶ The operators that are used to compare variables to check if they are similar or not. It will return value in true or false. These are also known as relational operators. comparison operators are:
- ▶ 1. > Greater than
- ▶ 2. < Less than
- ▶ 3. = Equal to

Assignment Operators

- ▶ The operator i.e used to assign values to identifiers, is known as assignment operator. There is only one assignment operator in C++. The assignment operator (=) is used to assign values to identifiers.

- ▶ Ex

a=2 [assign value 2 to a]



Bitwise Operator

- ▶ The operators which operate a bit level and allows the programmer to manipulate individual bits .These are basically used for testing or shifting bits.
- ▶ Ex

```
x<<3      // shift three bit position to left
```



Scope Resolution Operator (::)

- ▶ Like C,C++ is also a block-structured language .A variable declared in a block is said to be local to that block. In C, the global version of a variable cannot be accessed from within the inner block. C++ solves this problem by introducing the new operator called the scope resolution operator. This operator allows access to the global version of a variable.

Example to use the Scope Resolution Operator

```
• // Program to demonstrate the use of scope resolution operator
#include<iostream.h>
#include<conio.h>
int x = 10; // global x
int main()
{
    int x = 20; // x redeclared, local to main
    {
        int k = x;
        int x = 30; // x is declared again in inner block
        clrscr();
        cout << "\n We are in inner block \n";
        cout << "k = " << k << "\n";
        cout << "x = " << x << "\n";
        cout << ":: x = " << :: x << "\n";
    }
    cout << "\n We are in outer block \n";
    cout << "x = " << x << "\n";
    cout << ":: x = " << :: x << "\n";
    return 0;
}
```

Output

- We are in inner block

$k = 20$

$x = 30$

$\therefore x = 10$

- We are in outer block

$x = 20$

$\therefore x = 10$



THANK YOU