

FAST

**National University of Computer and
Emerging Sciences Peshawar**

OOP Lab # 2.1

DEPARTMENT OF COMPUTER SCIENCE

C++ Programming

Computer Instructor: Muhammad Abdullah Orakzai



Prepared By: Muhammad Abdullah Orakzai (Lab Instructor CS)

Contents



- 1) Operators in C++
- 2) Unary operators
- 3) Binary operators
- 4) Ternary operators
- 5) Lab Task
- 6) Home Tasks



Operators

- ❖ Operator is a symbol which is used to perform some operation.
- ❖ Operators are used to perform operations on variables and values.
- ❖ In the example below, we use the + **operator** to add together two values:
- ❖ `int x = 100 + 50;`
- ❖ Although the + operator is often used to add together two values, like in the example above, it can also be used to add together a variable and a value, or a variable and another variable:



Operators...

Example

```
int sum1 = 100 + 50;    // 150 (100 + 50)
int sum2 = sum1 + 250;   // 400 (150 + 250)
int sum3 = sum2 + sum2;   // 800 (400 + 400)
```



Types of Operators

1. Unary operators
2. Binary operators
3. Ternary operators



1) Unary Operator

1. Increment (++)
2. Decrement (--)
3. Negation (!)



2) Binary Operator

1. Arithmetic (+, -, *, /, %)
2. Relational (>, <, >=, <=, !=, ==)
3. Logical (&&, ||)
4. Assignment (=)
5. Arithmetic Assignment operator (+=, -=, *=, /=, %=)



3) Ternary Operator

❖ Conditional operator (?:)

Example

(condition) ? statement 1 : statement 2;

int result= (n1>n2) ? n1 : n2;



3) Ternary Operator...

```
#include<iostream>
```

```
using namespace std;
```

Output: Good evening.

```
int main()
```

```
{
```

```
    int time = 20;
```

```
    string result = (time < 18) ? "Good day." : "Good evening.";
```

```
    cout << result;
```

```
    return 0;
```

```
}
```



Arithmetic Operators in C++

Arithmetic operators are used to perform common mathematical operations.

Operator	Name	Description	Example
+	Addition	Adds together two values	$x + y$
-	Subtraction	Subtracts one value from another	$x - y$
*	Multiplication	Multiplies two values	$x * y$
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	$x \% y$



Adding two integers

```
#include<iostream>

using namespace std;

int main()
{
    int n1, n2, sum;
    cout<<"Enter first number:\t";
    cin>>n1;
    cout<<"Enter 2nd number:\t";
    cin>>n2;
    sum=n1+n2;
    cout<<"The sum is:\t"<<sum<<endl;

}
```

```
Enter first number: 3
Enter 2nd number: 6
The sum is: 9
PS E:\FAST NUCES Dschawan\Muhammad
```



Assignment Operator

Assignment operators are used to assign values to variables.

In the example below, we use the **assignment** operator (**=**) to assign the value **10** to a variable called **x**:

Example

```
int x = 10;
```



Arithmetic Assignment Operator

The **addition assignment** operator (**+=**) adds a value to a variable:

Example

```
int x = 10;
```

```
x += 5;
```

Arithmetic Assignment Operator...

A list of all arithmetic assignment operators:

Operator	Example	Same As
=	$x = 5$	$x = 5$
+=	$x += 3$	$x = x + 3$
-=	$x -= 3$	$x = x - 3$
*=	$x *= 3$	$x = x * 3$
/=	$x /= 3$	$x = x / 3$
%=	$x \% = 3$	$x = x \% 3$



Relational/Comparison Operators

- ❖ Comparison operators are used to compare two values.
- ❖ **Note:** The return value of a comparison is either true (1) or false (0).
- ❖ In the following example, we use the **greater than** operator ($>$) to find out if 5 is greater than 3:

Example

```
int x = 5;
```

```
int y = 3;
```

```
cout << (x > y); // returns 1 (true) because 5 is greater than 3
```

Relational/Comparison Operators

A list of all relational operators:

Operator	Name	Example
==	Equal to	$x == y$
!=	Not equal	$x != y$
>	Greater than	$x > y$
<	Less than	$x < y$
>=	Greater than or equal to	$x >= y$
<=	Less than or equal to	$x <= y$

Logical Operators

Logical operators are used to determine the logic between variables or values:

Operator	Name	Description	Example
&&	Logical and	Returns true if both statements are true	<code>x < 5 && x < 10</code>
	Logical or	Returns true if one of the statements is true	<code>x < 5 x < 4</code>
!	Logical not	Reverse the result, returns false if the result is true	<code>!(x < 5 && x < 10)</code>



Increment and Decrement Operators

1) Increment Operator:

The operators that is used to add 1 to the value of a variable is called increment operator.

2) Decrement Operator :

The operator that is used to subtract 1 from the value of a variable is called decrement operator.



1) The Increment Operator (++)

- ❖ The increment operator is represented by a double plus (++) sign.
- ❖ It is used to add 1 to the value of an integer variable.
- ❖ This variable can be used before or after the variable name.
- ❖ For example, to add 1 to a value of variable `xy`, it is normally written as

`xy = xy + 1;`

- ❖ By using increment operator “++” it is written as

`xy++`



1) The Increment Operator (++)...

- ❖ The increment operator can be written either before or after the variable.
- ❖ If it is written before the variable, it is known as **prefixing**.
- ❖ If it is written after the variable, it is known as **post fixing**.
- ❖ Prefix and postfix operators have different effects when they are used in expressions.



i) Prefix Increment Operator

❖ When an increment operator is used in prefix mode in an expression, it adds 1 to the value of the variable **before** the values of the variable is used in the expression.



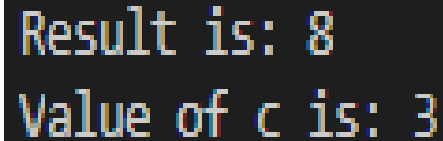
i) Prefix Increment Operator...

```
#include<iostream>

using namespace std;

int main()
{
    int a=2;
    int b=3;
    int c=2;
    int result=a+b(++c);
    cout<<"Result is: "<<result;
    cout<<"\nValue of c is: "<<c;

}
```

A screenshot of a terminal window showing the output of the C++ program. The text is displayed in a monospaced font with a light blue background. The output consists of two lines: "Result is: 8" and "Value of c is: 3".

Result is: 8
Value of c is: 3



i) Prefix Increment Operator...

- ❖ In the above program, 1 will be added to the value of **c** before it is used in the expression.
- ❖ Thus after execution, the result will be equal to 8 and the value of **c** will be 3.



ii) Postfix Increment Operator

❖ When an increment operator is used in postfix mode in an expression, it adds 1 to the value of the variable **after** the value of the variable is used in the expression.

❖ **For Example**, if in the above example, increment operator is used in postfix mode, the result will be different. The statement will be shown below:

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




ii) Postfix Increment Operator...

In this case, 1 will be added to the value of c after its existing value has been used in the expression. Thus after execution, the result will be equal to 7 and the value of c will be 3.

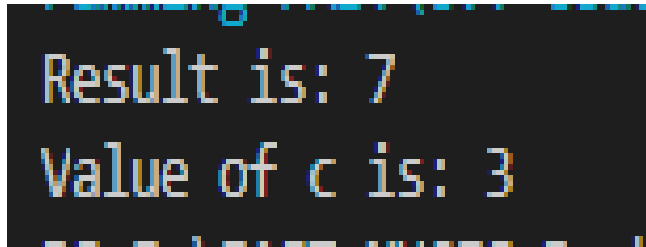
ii) Postfix Increment Operator...

```
#include<iostream>

using namespace std;

int main()
{
    int a=2;
    int b=3;
    int c=2;
    int result=a+b+(c++);
    cout<<"Result is: "<<result;
    cout<<"\nValue of c is: "<<c;

}
```



```
Result is: 7
Value of c is: 3
```



2) The Decrement Operator (--)

- ❖ The decrement operator is represented by a double minus (--) sign.
- ❖ It is used to subtract 1 from the value of an integer variable.
- ❖ This variable can be used before or after the variable name.
- ❖ For example, to subtract 1 from the value of variable `xy`, the decrement statement is written as

`xy--;` or `--xy;`



i) Prefix Decrement Operator

❖ When decrement operator is used in prefix mode in an expression, it subtracts 1 from the value of the variable **before** the values of the variable is used in the expression.



i) Prefix Decrement Operator...

```
#include<iostream>

using namespace std;

int main()
{
    int a=2;
    int b=3;
    int c=2;
    int result=a+b+(--c);
    cout<<"Result is: "<<result;
    cout<<"\nValue of c is: "<<c;

}
```

A screenshot of a terminal window with a black background and white text. It displays the output of the C++ program: "Result is: 6" on the first line and "Value of c is: 1" on the second line.

Result is: 6
Value of c is: 1



i)Prefix Decrement Operator...

- ❖ In the above program, 1 will be subtracted from the value of **c** before it is used in the expression.
- ❖ Thus after execution, the result will be equal to 6 and the value of **c** will be 1.



ii) Postfix Decrement Operator

❖ When an decrement operator is used in postfix mode in an expression, it subtracts 1 from the value of the variable **after** the values of the variable is used in the expression.

❖ For Example, if in the above example, decrement operator is used in postfix mode, the result will be different. The statement will be shown below:

```
result =a + b + c--;
```



ii) Postfix Decrement Operator...

In this case, 1 will be subtracted from the value of `c` after its existing value has been used in the expression. Thus after execution, the result will be equal to 7 and the value of `c` will be 1.

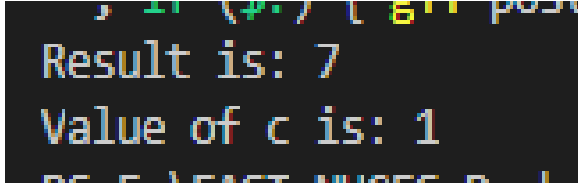
ii) Postfix Decrement Operator...

```
#include<iostream>

using namespace std;

int main()
{
    int a=2;
    int b=3;
    int c=2;
    int result=a+b+(c--);
    cout<<"Result is: "<<result;
    cout<<"\nValue of c is: "<<c;

}
```



```
Result is: 7
Value of c is: 1
```



Class Task-1

Ask user to enter a three digit number and then display the number in reverse order.

Task-1 Solution

```
#include<iostream>

using namespace std;

int main()
{
    int number;
    cout<<"Enter 3 digit number:";
    cin>>number;
    cout<<number%10;
    number=number/10;
    cout<<number%10;
    number=number/10;
    cout<<number;

}
```

```
Enter 3 digit number:123
321
```

```
Enter 3 digit number:456
654
```



Assignment # 01

- 1) Write a C++ program that will convert dollar to rupees (Dollar to Rupees Conversion Calculator).
- 2) Write a C++ program that will convert rupees to dollar (Rupees to Dollar Conversion Calculator).
- 3) Write a C++ program that will convert centigrade to Fahrenheit.
- 4) Take student name and marks of your 2nd semester from user and then generate DMC which will contain obtained marks out of total and percentage.



References

- <https://beginnersbook.com/2017/08/cpp-data-types/>
- <https://www.geeksforgeeks.org/c-data-types/>
- http://www.cplusplus.com/doc/tutorial/basic_io/
- <https://www.geeksforgeeks.org/basic-input-output-c/>
- <https://www.w3schools.com/cpp/default.asp>
- <https://www.javatpoint.com/cpp-tutorial>

THANK YOU

