**C++ Operators Programs**

## Arithmetic Operator

They are five arithmetic operators in C++.

* + Addition or unary plus
* - Subtraction or unary minus
* \* Multiplication
* / Division
* % Modulo operator

These operators can operate on any arithmetic operations in C++.

## Example Program Of Arithmetic Operators

// Header Files

#include<iostream>

#include<conio.h>

//Main Function

using namespace std;

int main ()

{

// Variable Declaration

int a = 200;

int b = 26;

int c = 50;

int d = 40;

int result;

cout << "Simple Arithmetic Operators Example Program \n";

result = a - b; // subtraction ( Subtraction or unary minus Arithmetic Operator)

cout <<"\na - b = "<<result;

result = b \* c; // multiplication ( Multiplication Arithmetic Operator)

cout <<"\nb \* c = "<< result;

result = a / c; // division ( Division Arithmetic Operator)

cout <<"\na / c = "<< result;

result = a + b \* c; // precedence ( Addition or unary plus Arithmetic Operator)

cout <<"\na + b \* c = "<< result;

cout <<"\na \* b + c \* d = "<< a \* b + c \* d; // Mixed

return 0;

}

## Sample Output

Simple Arithmetic Operators Example Program

a - b = 174

b \* c = 1300

a / c = 4

a + b \* c = 1500

a \* b + c \* d = 7200

--------------------------------

Process exited after 1.648 seconds with return value 0

Press any key to continue . . .

## Logical Operators

* AND, OR operators are used when we want to use two or more Conditions.

## Types OfLogical Operators

* && Logical AND
* || Logical OR
* ! Logical NOT

## Logical And (&&) Operator

### Logical And Operator Definition

If both the operations are successful, then the condition becomes true.

### Logical And Operator Syntax

expr1 && expr2

### Logical And Operator Syntax Example

if( (a>10) && (a<20) )

printf(?A is in-between of 10 and 20?);

### Logical And Operator Example Program

// Header Files

#include<iostream>

#include<conio.h>

//Main Function

using namespace std;

int main ()

{

// Variable Declaration

int num1 = 30,num2 = 40;

//int num1 = 50,num2 = 80;

cout << "Simple Logical Operators Example Program \n";

if(num1>=40 && num2>=40){

printf("Num 1 and Num 2, both are greater than or equal to 40");

}

if(num1>=40 || num2>=40){

printf("Num 1 or Num 2 is greater than or equal to 40");

}

// Wait For Output Screen

getch();

//Main Function return Statement

return 0;

}

## Sample Output

Simple Logical Operators Example Program

//if num1 = 30,num2 = 40

Num 1 or Num 2 is greater than or equal to 40

//int num1 = 50,num2 = 80;

Num 1 and Num 2, both are greater than or equal to 40

## Unary Operators

There are two Unary Operators. They are Increment and Decrement.

## Increment Unary Operator

variable++

++variable;

Is Equivalent i=i+1 or i+=1

## Increment Unary Operator Types

* Post Increment i++
* Pre Increment ++i

## Decrement Unary Operator

variable--;

--variable;

Is Equivalent i=i-1 or i-=1

## Decrement Unary Operator Types

* Post Decrement i--
* Pre Decrement --i

## Unary Operators Explanation

* ++i: increments l and then uses its value as the value of the expression;
* i++: uses l as the value of the expression and then increments l;
* --i: decrements l and then uses its value as the value of the expression;
* i--: uses l as the value of the expression and then decrements l.
* Change their original value.

## Example Program For Unary Operators

// Header Files

#include<iostream>

#include<conio.h>

//Main Function

using namespace std;

int main ()

{

// Variable Declaration

int a = 10;

cout << "Simple Unary Operators Example Program \n";

/\* Increment Operators \*/

cout<<"\nPost Increment = "<<a++; //Post Increment

a = 10;

cout<<"\nPre Increment = "<<++a; //Pre Increment

/\* Decrement Operators \*/

a = 10;

cout<<"\nPost Decrement = "<<a--; //Post Decrement i--

a = 10;

cout<<"\nPre Decrement = "<<--a; //Pre Decrement i--

// Wait For Output Screen

getch();

//Main Function return Statement

return 0;

}

## Sample Output

Simple Unary Operators Example Program

Post Increment = 10

Pre Increment = 11

Post Decrement = 10

Pre Decrement = 9

## Definition

Check condition if true, it returns first variables value otherwise return second values. sometimes it replaces if..else statement

## Syntax

Condition? Expression1: Expression2

## Example

(a>10) ? b : c

### Explanation For Conditional or Ternary operator

Given that

a, b, c

are expressions;

the expression

(a>10) ? b : c

has as its value b if a is nonzero, and c otherwise. Only expression b or c is evaluated.

Expressions b and c must be of the same data type. If they are not but are both arithmetic data types, the usual arithmetic conversions are applied to make their types the same. It is also called ternary operators.

## Example Program For Conditional or Ternary operator

// Header Files

#include<iostream>

#include<conio.h>

//Main Function

using namespace std;

int main ()

{

// Variable Declaration

int a = 10;

int b = 15;

int c;

cout << "Simple Conditional or Ternary Operators Example Program \n";

c = a <= b ? a : b; //Conditional or Ternary

cout<<"\nC Is "<<c;

// Wait For Output Screen

getch();

//Main Function return Statement

return 0;

}

## Sample Output

Simple Conditional or Ternary Operators Example Program

C Is 10

## Definition

The scope resolution operator is used for the Unary scope operator if a namespace scope  (or) Global Scope

## Scope Resolution Operator Syntax

:: identifier // for Global Scope

class-name :: identifier // for Class Scope

namespace :: identifier // for Namespace Scope

//simple syntax

:: global variable name

## Scope Resolution Operator Example Program

// Header Files

#include<iostream>

#include<conio.h>

#include<stdio.h>

//Main Function

using namespace std;

int a = 100; //Global variable

int main ()

{

// Local Variable Declaration

int a = 200;

cout << "Simple Scope Resolution Example Program In C++\n";

// Print Global Varibale

cout << ::a<< endl;

// Print Local Varibale

cout << a << endl;

//Main Function return Statement

return 0;

}

## Sample Output

Simple Scope Resolution Example Program In C++

100

200