

➤ Increment operator

The increment operator is used to **increase** the value of a variable by 1. it is denoted by the symbol **++**.it is a unary operator and works with single variable.

The increment operator cannot increment the value of contents and expression. For example **A ++** and **X ++** are **valid** statements, but **10 ++** is an **invalid** statement. Increment operator can be used in tow forms.

- **Prefix Form**
- **Postfix Form**

Prefix form

In prefix form, the increment operator is written **before** the variable.

Example

++X

The above line **increments** the value of variable **X** by **1**.

Postfix form

In postfix form, the increment operator is written **after** the variable.

Example

X++

The above line **increments** the value of variable **X** by **1**.

Different between prefix and postfix increment

When increment operator is used independently, **prefix** and **postfix** from work similarly, for example the result of **A ++** and **++ A** is same. But when increment operator is used in a **larger expression** with other operators, **prefix** and **postfix** forms work differently.

Example

The result of two statements **A=++B** and **A = B++** are different.

The statement **A = ++ B** contains two operators **++** and **=**.

1. It increase the value of **B** by **1**.
2. It assign the value of **B** to **A**.

Program

Write a program that explains the difference of postfix increment operator and prefix increment operator used as part of a larger expression.

```
#include<iostream>
Using namespace std;
Main()
{
Int a,b,x,y;
a = b = x = y = 0;
b = a ++;    // postfix
y = ++ x;    // prefix
cout<<" a = " << a << endl << " b = " << b << endl;
cout<<" x = " << x << endl << " y = " << y << endl;
}
```

OutPut

a = 1
b = 0
x = 1
y = 1

➤ Decrement operator

The decrement operator is used to decrease the value of a variable by 1. it is denoted by the symbol -- .it is a unary operator and works with single variable.

The decrement operator cannot i decrement the value of contents and expression. For example **A --** and **X --** are **valid** statements, but **10 --** is an **invalid** statement. Decrement operator can be used in tow forms.

- Prefix Form
- Postfix Form

Prefix form

In prefix form, the decrement operator is written **before** the variable.

Example

--X

The above line **decrement** the value of variable **X** by **1**.

Postfix form

In postfix form, the **decrement** operator is written **after** the variable.

Example

X--

The above line **decrement** the value of variable **X** by **1**.

Different between prefix and postfix Decrement

When decrement operator is used independently, **prefix** and **postfix** from work similarly, for example the result of **A --** and **-- A** is same. But when decrement operator is used in a larger expression with other operators, **prefix** and **postfix** forms work differently.

Example

The result of two statements **A=--B** and **A = B--** are different.

The statement **A = -- B** contains two operators **--** and **=**.

3. It decrements the value of **B** by **1**.
4. It assign the value of **B** to **A**.

Program

Write a program that explains the difference of postfix decrement operator and prefix decrement operator used as independent expression.

```
#include<iostream>
using namespace std;
main()
{
int a,b,x,y;
a = b = x = y = 0;
a--; // postfix
b = a;
-- x; // prefix
y = x;
cout<<" a = " << a << endl << " b = " << b << endl;
cout<<" x = " << x << endl << " y = " << y << endl;
}
```

OutPut

```
a = -1
b = -1
x = -1
y = -1
```

➤ Type Casting

The process of converting the data type of a value during execution is known as **type casting**. Type casting can be performed in two ways.

- **Implicit type casting**
- **Explicit type casting**

Implicit type casting

Implicit type casting is performed automatically by the C++ compiler.

Example

```
Int a = 3.56;
```

```
Cout<< a;
```

Now the value of **a** is **3**.

Explicit type casting

Explicit type casting is performed by programmer.

Syntax

(Type) expression;

Example

```
float a = 3.56;
```

```
Cout<< (int)a;
```

Now the value of **a** is **3**.