

## ★ TYPE CONVERSION

\* When an operation is performed on the operands with different data type, the data type of one operand will be converted to another and its called type conversion.

\* Type conversion can be done in 2 ways.

1) Implicit type conversion (Type promotion)

2) Explicit type conversion (Type casting)

### 1) Implicit type conversion

\* When an expression involves different data types c++ converts the lower sized operand to the type of highest sized operand and its called as implicit type conversion or type promotion.

Eg :-

```
#include <iostream.h>
```

```
void main()
```

```
{
```

```
int a, b;
```

```
float c;
```

```
cout << "enter numbers";
```

```
cin >> a >> b;
```

```
c = a + b; /* type promotion will be applied by converting int to float  
as float is the highest compared to int data type, output will  
be a floating point value. */
```

```
cout << c; }
```

### 2) Explicit type conversion

\* In this type conversion programmer will manually perform the conversion. i.e. the data type of the result of operation will be specified by the programmer.

syntax

(type) expression

Eg :- int x, y, z;

float c;

c = (float) x + y / z;

\* The result of the operation will be converted to float before storing in to the variable 'c'.

## ★ STATEMENTS IN C++

- \* Smallest executable unit of a programming language.
- \* Uses semi colon (;) as delimiter for the statement.
- \* Different types of statements in c++ includes.
  1. Declaration statements
  2. Assignment statements
  3. Input statements.
  4. Control statements.

### 1. DECLARATION STATEMENT

- \* variables should be declared before it is used in the program.
- \* variables are user's defined word and its considered as an identifier.

Syntax

Data-type <variable 1>, <variable 2>, <variable 3> .....

- \* Each variables should be separated by comma (,).

### VARIABLE INITIALISATION

- \* Assigning a value to variable during variable declaration is variable initialization.

Eg:- `int a=5, b=20;`

- \* value can be stored in the variable either during compilation or during the execution of the program.

const - The access modifier

- \* The keyword `const` is used to make a variable to a constant value. i.e the value of such variables cannot be changed during the execution.

Eg:- `const float p=3.14;`

- \* The above variable declaration the value of variable 'p' is initialized to 3.14 with the keyword `const` as a result the value of variable 'p' cannot be changed through out the program.

## 2. ASSIGNMENT STATEMENT

Assignment operator is used to assign values to a variable.

Eg :-

A = 15

$x = \text{sqrt}(a)$  → parameter/argument

c = a + b

c = a

## TYPE COMPATIBILITY

The 2 different possibilities that occurs when an assignment statement is executed.

1. Size of the data type of the variable at LHS is higher than the variable or expression on the RHS. In this case data type of the value at the RHS is promoted (type promotion) to that of the variable at LHS.

```
#include <iostream.h>
```

```
void main()
```

```
{
```

```
int a=5, b=6;
```

```
float c;
```

```
c = a;
```

```
cout << c;
```

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

```
cout << c;
```

```
}
```

output

5.0 11.0

2. Size of the data type of the variable at LHS is smaller than the size of RHS value.

```
#include <iostream.h>
```

```
void main()
```

```
{
```

```
float a=5.3, b=3.4;
```

```
int c;
```

```
c = a;
```

```
cout << c;
```

```
c = a + b
```

```
cout << c;
```

```
}
```

output

58

### 3. INPUT STATEMENTS

- \* The keyword cin followed by getform or extraction operator (>>) specifies the input operation.

Eg:-

```
cin >> a >> b;
```

### 4. OUTPUT STATEMENTS

- \* The keyword cout followed by put to or insertion operator (<<) specifies the output operation.

Eg:-

```
cout << a << b;
```

### CASCADING OF I/O OPERATORS

- \* The statement

```
cin >> a;
```

```
cin >> b;
```

```
cin >> c;
```

The above 3 statements can be combined to and rewritten into a single statement.

```
cin >> a >> b >> c; and this called cascading of I/O operators.
```