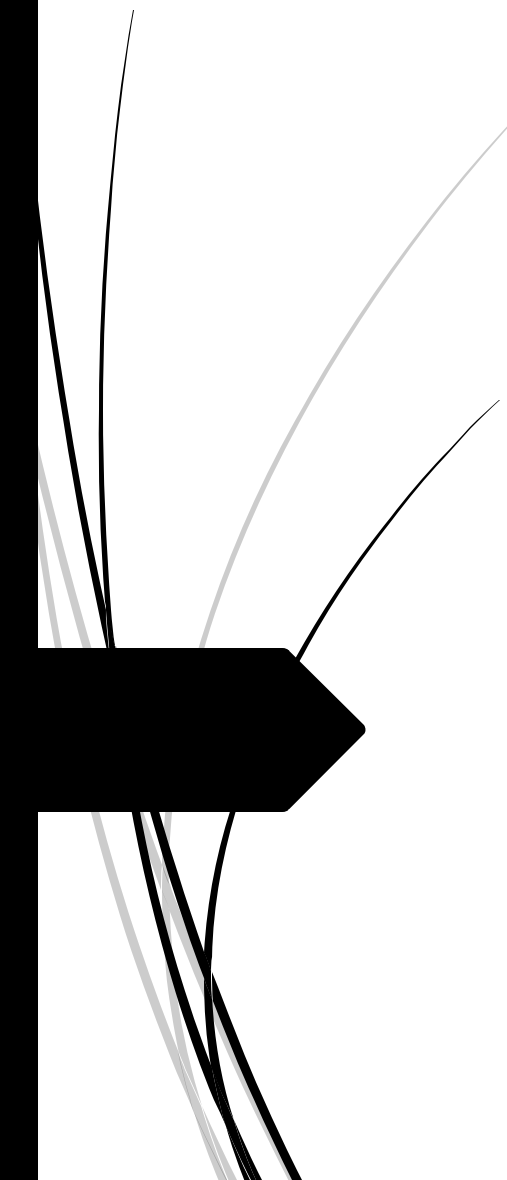


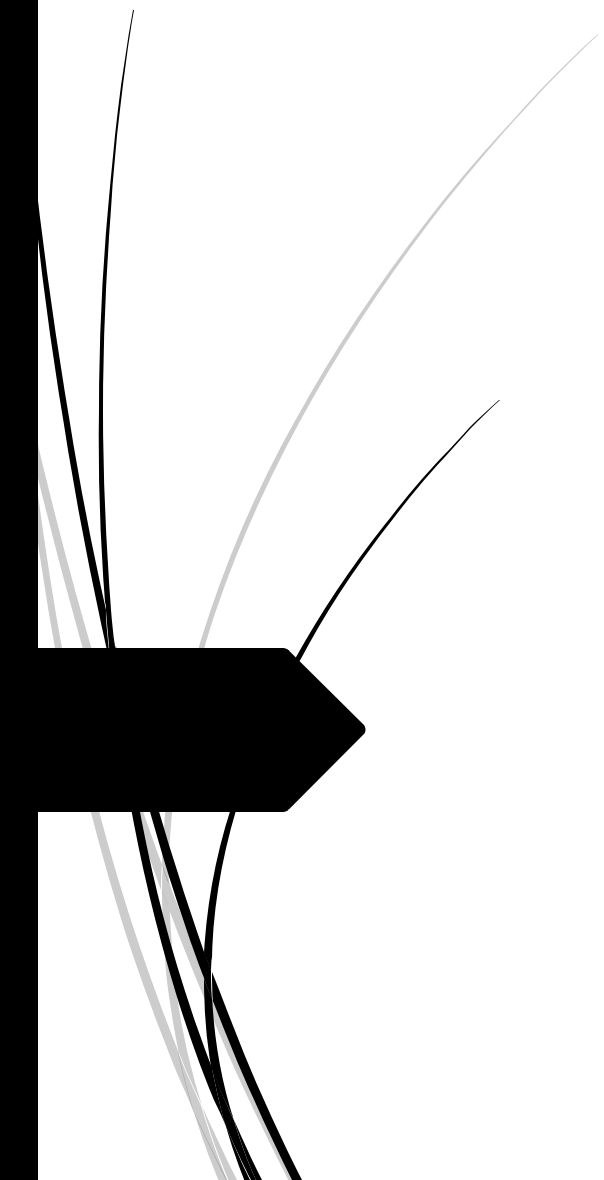
Data Type and Operators

Introduction

- All the data are stored in the form of basic memory units called bits.
- A bit is a unit of memory that has only two possible states 0 & 1
- The smallest memory unit is 1 byte. (8 bits)

Data Type

- **The data type defines which kinds of data will be stored in variables**
- **And also define how many memory used for the data**



Data Type

C++ Data types

Data type is used to tell the computer

→ **Type of data**

→ **Size of memory used for store the data**

- **Classified into 3**

1) **Built in data type (Fundamental Data type)**
2) **Derived data type**
3) **User defined data type**

Data types

1) **Built in data type**
   - **Char (1 byte)**
   - **Int (2 byte)**
   - **Float (4 byte)**
   - **Double (8 byte)**
   - **Void (0 byte)**
2) **Derived data type**
3) **User defined data type**

Data types

1) **Built in data type**
2) **Derived data type**
   - **Array**
   - **Pointer**
   - **Functions**
   - **reference**
3) **User defined data type**

Data types

1) **Built in data type**
2) **Derived data type**
3) **User defined data type**
   - **structure**
   - **union**
   - **class**
   - **enumeration**

Built-in / Fundamental Data type

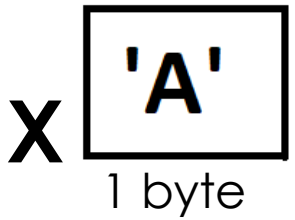
	Data type	Memory size
Character :	char	1 byte
Integer :	int	2 byte
Floating point :	float	4 byte
Double floating point :	double	8 byte
Valueless :	void	____

Character (char)

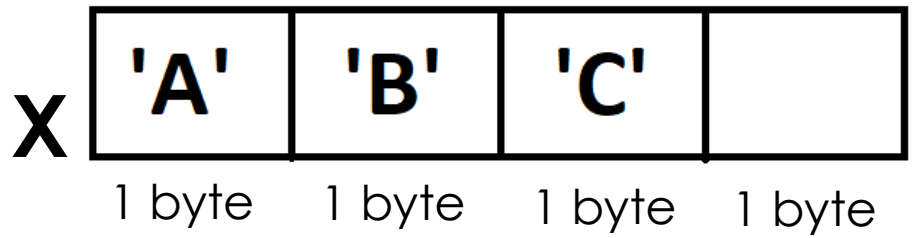
- **used to store a character value. (letters)**
- **Character literals are written in single quotes, like this: 'A'**
- **for multiple characters (strings) use double quotes: "ABC“**

Example:

char x='A';



char x="ABC";

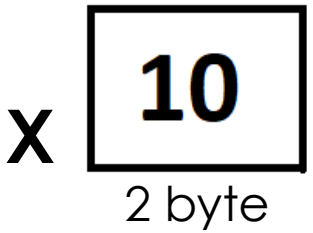


Integer (int)

- **used to store an integer value. (numbers)**
- **Not store fractional part**

Example:

int x=10;



int count = 0;

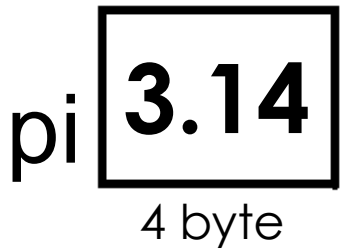
Floating point (float)

- **used to store an floating point value. (real numbers)**
- **store fractional part**

  Example:

float x=10.5;

float pi = 3.14;



Double

- → **used to store an floating point value. (real numbers)**
- → **store fractional part**
- → **Double memory of float type**

Example:

```
double x=10.5071;
```

x **10.5071**

8 byte

```
double mark = 20;
```

mark **20.0000**

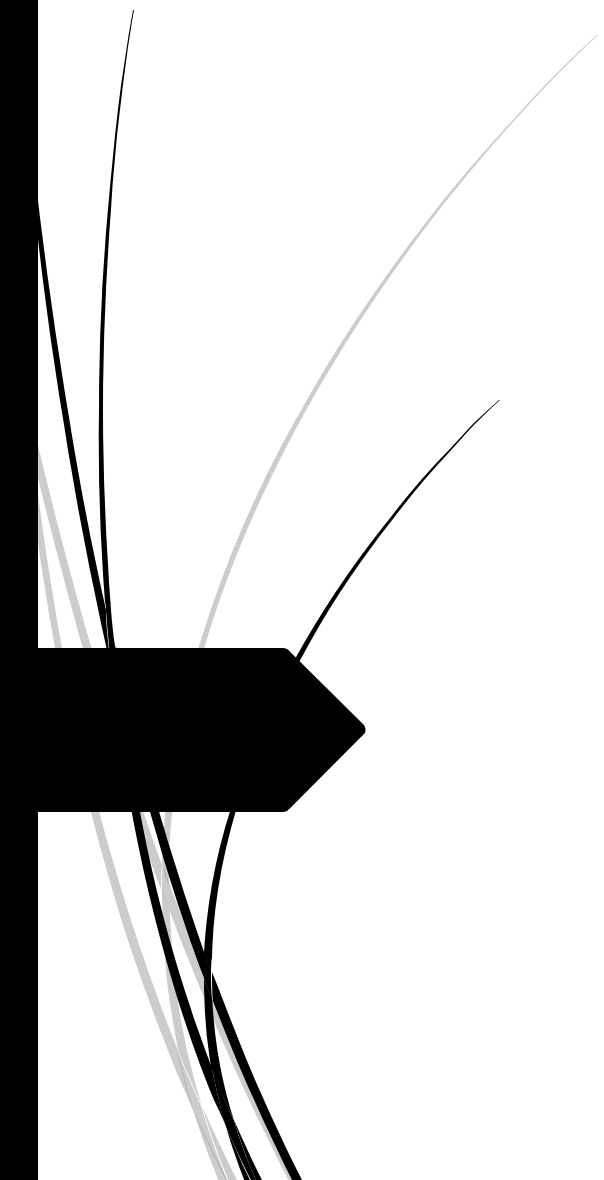
8 byte

Void

- **used only in function declarations**
- **Use zero byte memory**

Example:

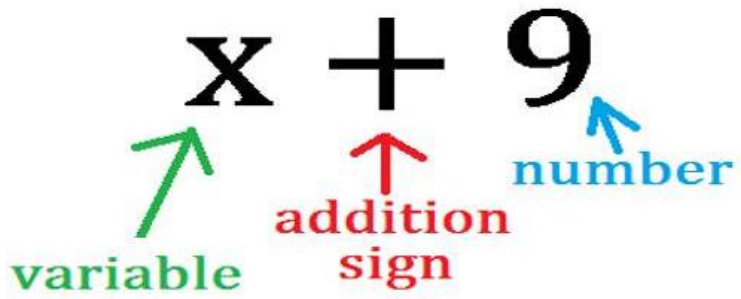
```
void main()
void read()
void display()
```

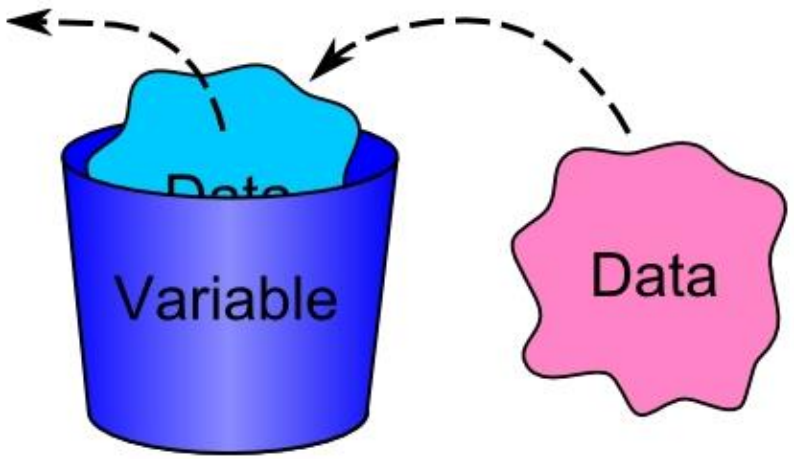


Variable,Constant,Expression

Identifiers (Variables)

- → A variable is a **letter or symbol** used to **represent a value** that can **change**
- → The value may be number or characters ..





Identifiers (Variables) naming rules

1. **Only alphabetic characters (A-Z), digits (0-9) and underscore (_)are permitted**
2. **Name cannot start with digit**
3. **Uppercase(A-Z) and lowercase(a-z) are distinct**
4. **Keywords cannot be used as variables**

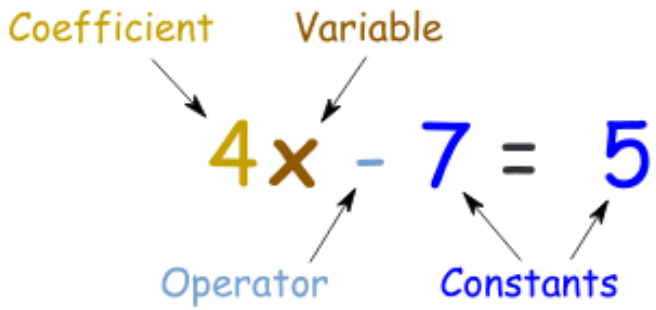
Example :

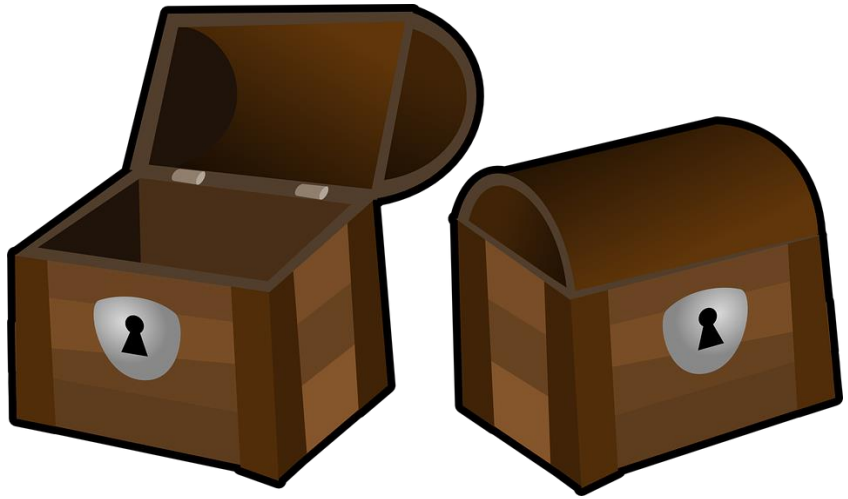
```
int a = 50;
float mark = 26.7;
void _display()
```

_Name
_Num5
num5

Constants

- → **A Constant is a value that does not change**
- → **Ex: 5,0,3.14,**



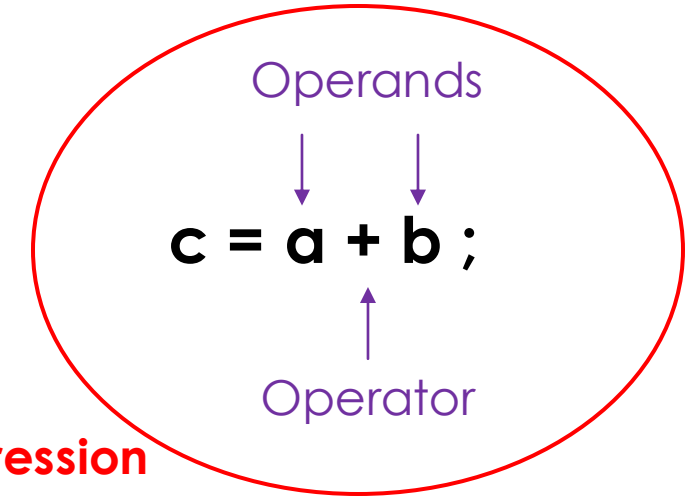


Expressions

- **Expression is a sequence of operators and their operands that specifies a process or computation**

Example :

```
c = a + b;
avg = sum / total_num ;
```

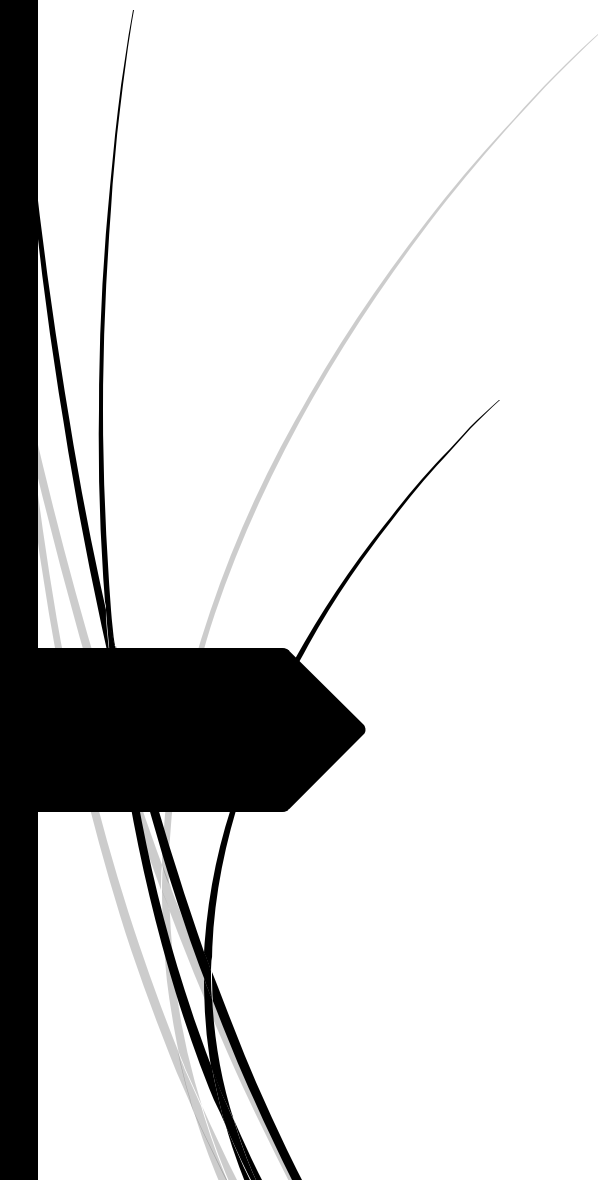


Statements

- **Smallest executable unit** of a program
- All statement end with **semicolon** (;)

- **Declaration** statement
  - Variables are must be declare before using it.
  - Example : int a;
    int x, y, z;

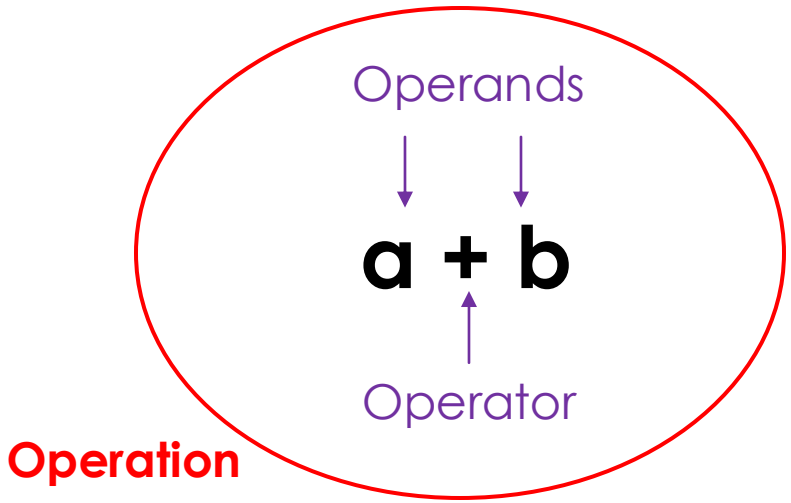
- **Assignment** statement
  - Assign the value of RHS to LHS
  - Example : Age = 18 ;
    int x = 10, y = 5 ;



Operators

Operators

- It is a symbol that performs an operation
- The meaning of the symbols are already defined in the C++ Compiler
- The data in an operation is called operands



Operators are…

1) **Input/Output Operators**
2) **Arithmetic Operators**
3) **Relational Operators**
4) **Logical Operators**
5) **Conditional Operators**
6) **Assignment Operators**
7) **Increment & decrement Operators**
8) **SizeOf Operators**

Input / Output Operators

- **Used to perform Input / output operations**

Operator	Description	Example
>>	Input	cout << "Hello world"
<<	Output	cin >> mark

Arithmetic Operators

Operator	Description	Example
+	Addition	x + y
–	Subtraction	x – y
*	Multiplication	x * y
/	Division	x / y
%	Modulo	x % y

Relational Operators

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

Logical Operators

Operator	Description	Example
&&	Logical AND	x && y
||	Logical OR	x || y
!	Logical NOT	!x

Logical NOT : Unary operator

Conditional Operators

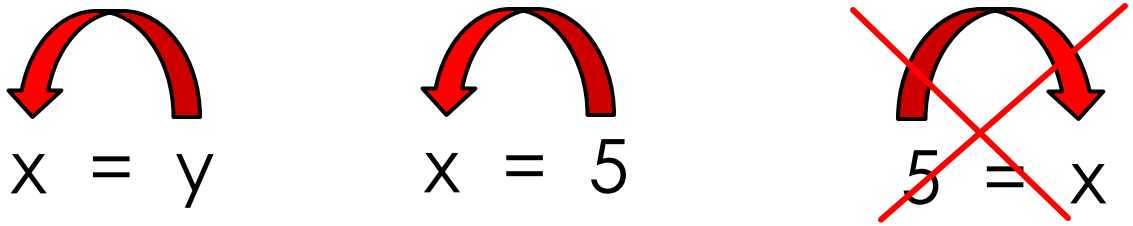
- **It is a ternary Operator**

Operator	Example
? :	(A == 10) **?** A++ **:** A-- ;

(N == 10) ?

cout << “Number is 10” : cout << “Number is not 10” ;

Assignment Operators



Operator	Description	Example
=	Assignment	x = y
+=	Add & assign	x += y
–=	Subtract & assign	x –= y
*=	Multiply & assign	x *= y
/=	Division & assign	x /= y

Increment & Decrement Operators

Operator	Description	Example
++	Increment	X ++
– –	Decrement	Y – –

Unary operator

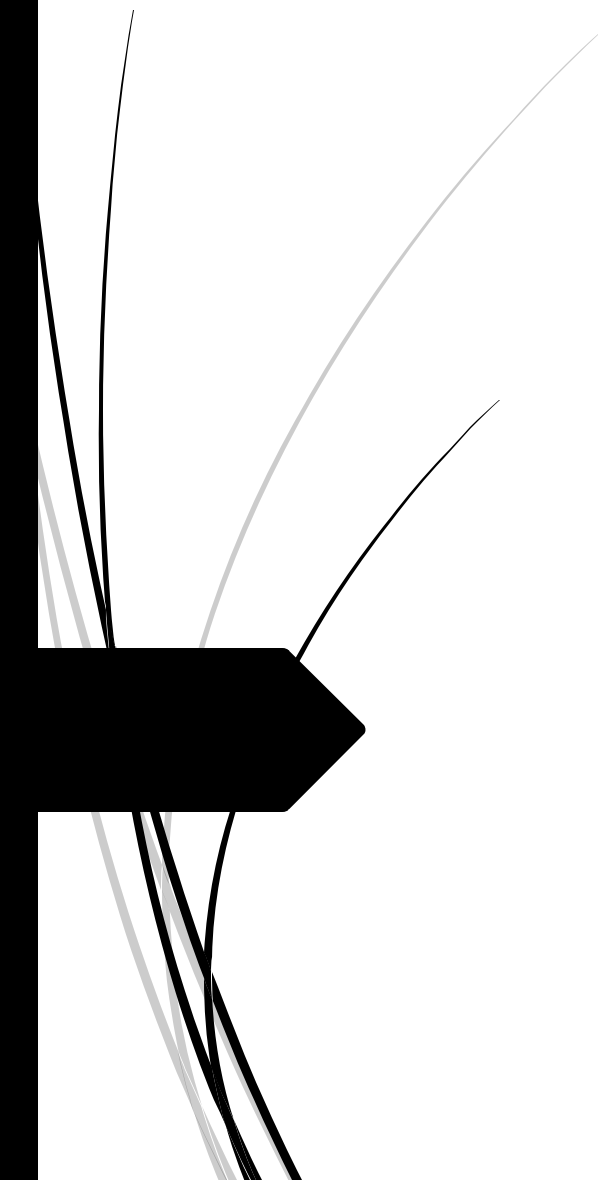
x = x + 1 x += 1 x ++

x = x – 1 x –= 1 x – –

SizeOf Operators

- **Used to find the size of a data type**
- **Example :**

```
sizeof(int);
sizeof (float)
```



Type Conversion

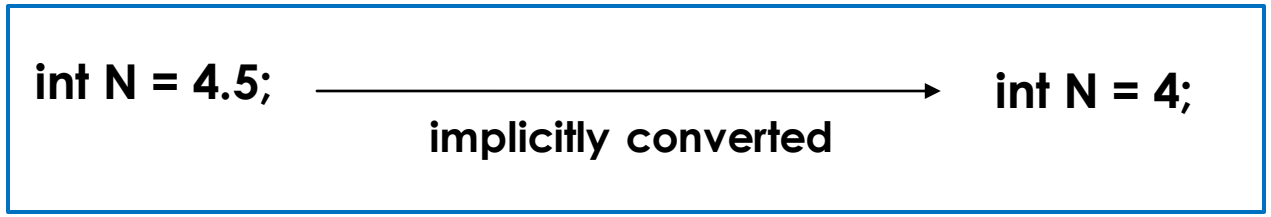
Type Conversion

- **Data are converted from one type to another type are called type conversion**

- **Classified into 2**
  1) **Implicit type conversion**
  2) **Explicit type conversion**

Implicit Type Conversion

- It is automatically done by the compiler.
- For example ,
  - float N = 5; → 5 is an integer type, float 5 is implicitly converted into 5.0000
  - int N = 6.54 ; → 6.54 is a float type, int 6.54 is implicitly converted into 5

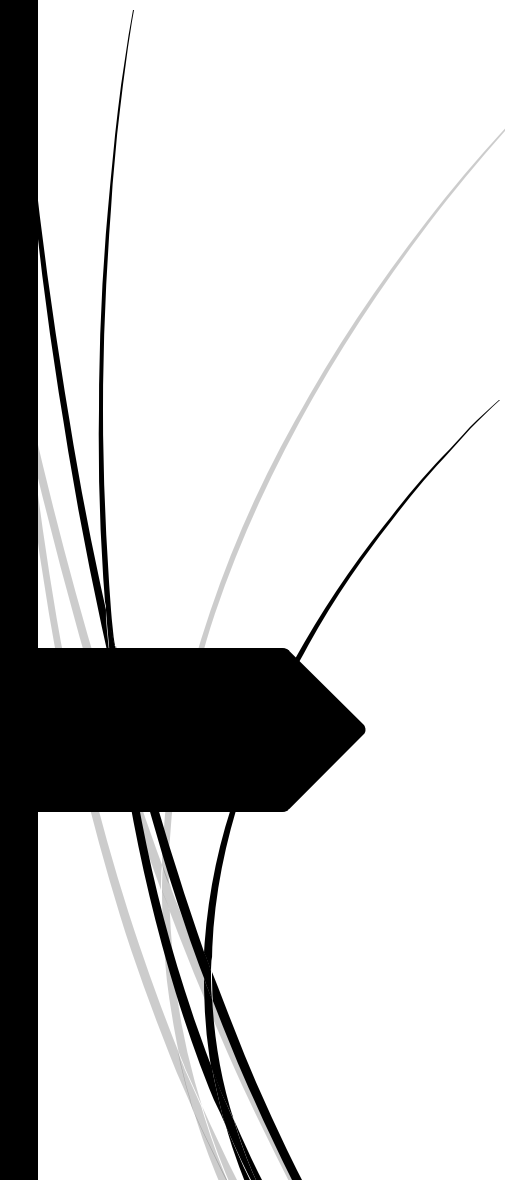


Explicit Type Conversion

- **It is done by the program code.**
- **For example ,**

```
int N = 10;
Int X = 4;
float S = (float) N / X
```

Data Type tell the compiler how to transfer it



Structure of C++ program

```
// Command line
# include < Header File >
using namespace std;
int main()
{
    ……..
    ……..
    ……..
}
```

Body of the program

Teaching materials Prepared by

RIDHUN DEV

Lecture IHRD College

+1 / +2 Computer Science

8089552581