+1 Computer Application/Science

Data Types and Operators

Short Note

To watch Video Class Click Here

Data types

To identify the type of data and associated operations

ഡാറ്റയുടെ തരവും അനുബന്ധ പ്രവർത്തനങ്ങളും തിരിച്ചറിയാൻ ഉപയോഗിക്കുന്നു

There three types of data types

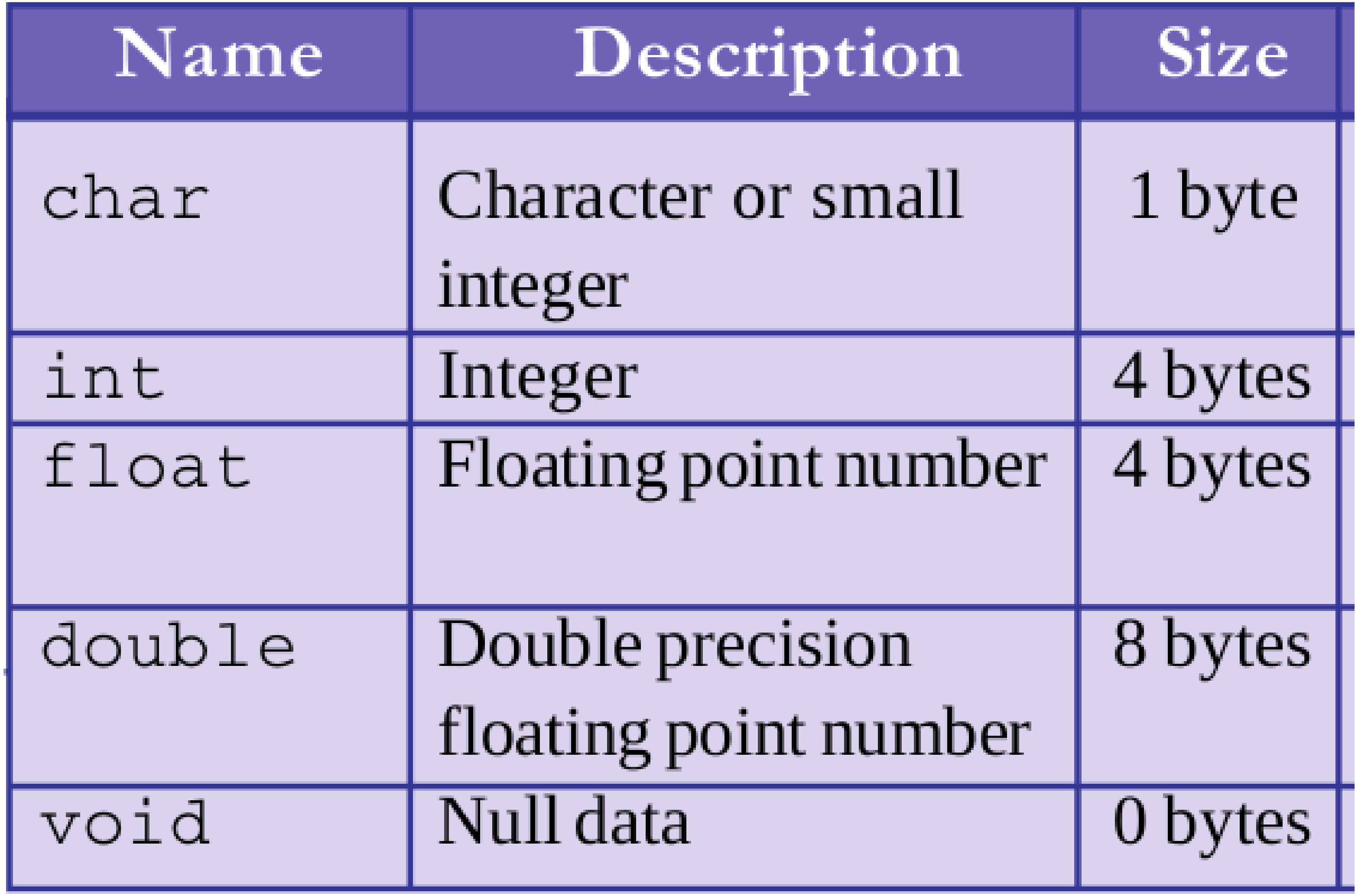
- Fundamental
- User defined
- Derived

Fundamental data types

Fundamental data types are defined in C++ compiler. They are also known as built-in data types.

The following are the fundamental data types

int	For representing integer number
float	For representing floating point numbers
double	For representing more precision floating point numbers
char	For representing a single character
void	For representing empty set of values



Name	Description	Size
`char`	Character or small integer	1 byte
`int`	Integer	4 bytes
`float`	Floating point number	4 bytes
`double`	Double precision floating point number	8 bytes
`void`	Null data	0 bytes

User-defined data types

Data type defined by the programmer is user defined data type.

Example: struct,class,union

Derived data types

Data type derived from fundamental data types is derived data types.

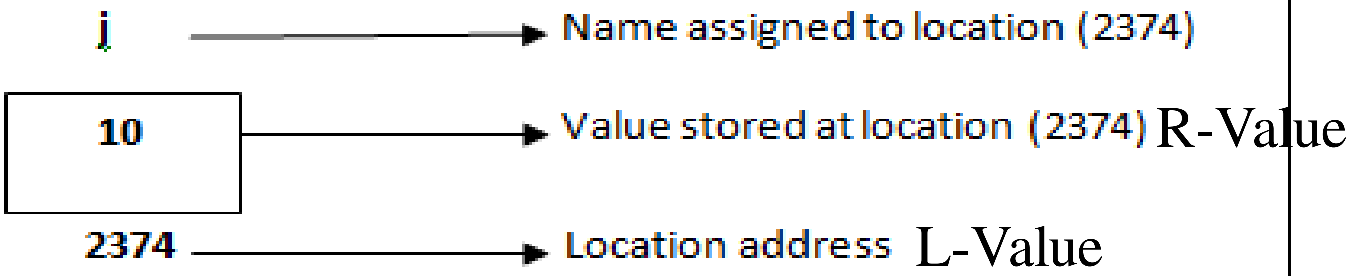
Example: Arrays, pointers, functions

Variables

Named memory location is called variables. There are three important aspects for a variable.

- Variable Name
- Memory address(L-Value)
- Content(R-Value)

int i =10


i
Name assigned to location (2374)
10
Value stored at location (2374)
R-Value
2374
Location address
L-Value

Operators

Is a symbol to represent an operation.

Example : In a+b , + is an operator

Based on number of operands operators are classified into

Unary:Which takes only one operand Example: Unary + ,- (+45,-12)

Binary:Which takes two operands Example: Arithmetic operators(+,-,*,/,%), Relational oprators(<,>,<=,>=,==,!=)...

Ternary:Which takes three operands. Example: Conditional operator(?:)

Based on the nature of operation operators are classified into

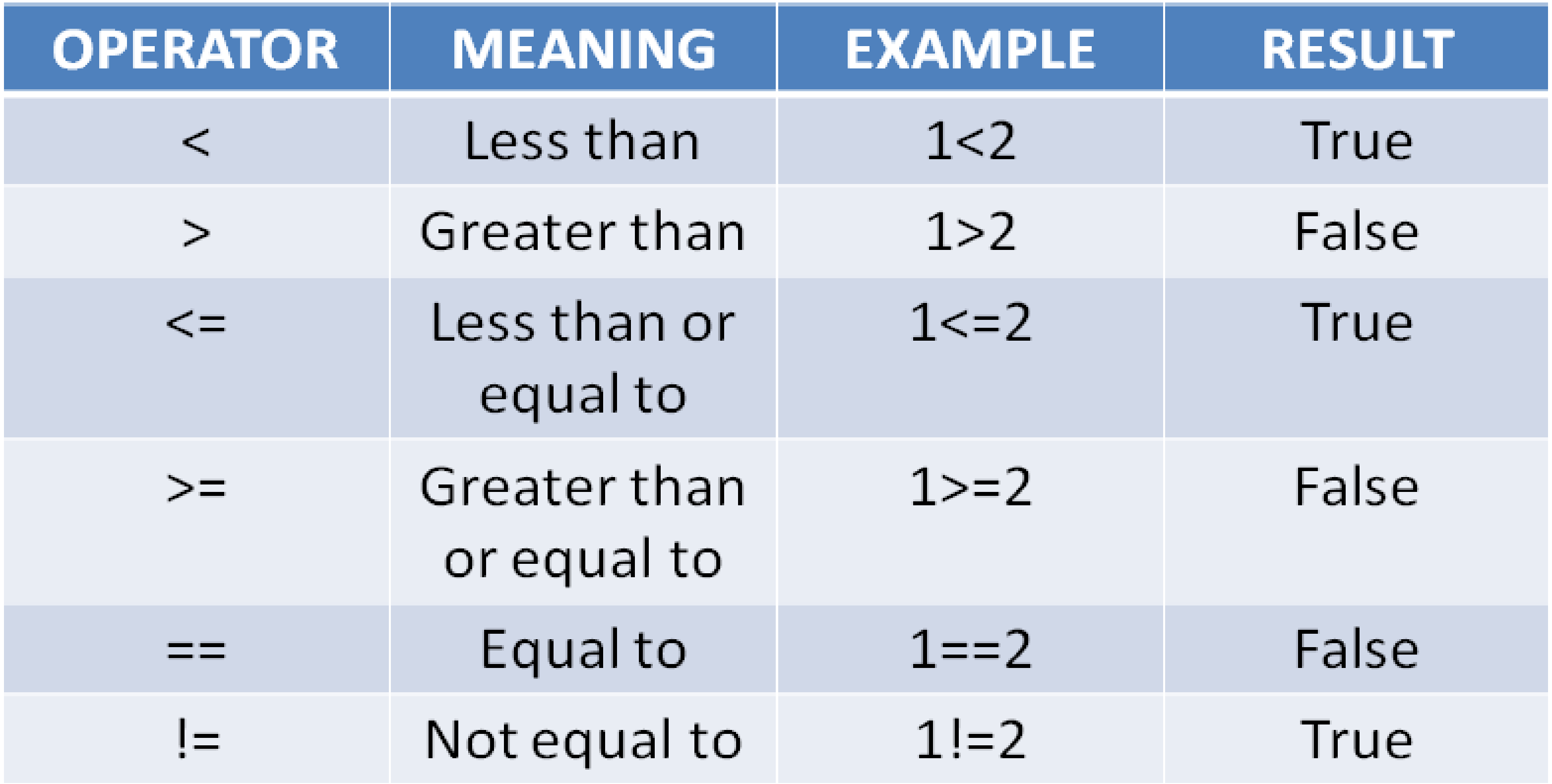
- Arithmetic operators   +,-,*,/,%
- Relational operators   <,>,<=,>=,==,!=
- Logical operators   !,&&,||
- Increment/Decrement   ++,--
- Assignment operator =
- Arithmetic Assignment +=,-=,*=,/=,%=
- Input/Output operators   >>,<<
- Conditional operator (Ternary operator) ?:

Arithmetic Operators

Operators	Meaning	Example	Result
+	Addition	4+2	6
-	Subtraction	4-2	2
*	Multiplication	4*2	8
/	Division	4/2	2
%	Modulus operator to get remainder in integer division	5%2	1

Relational Operators

Used for comparing numeric values. Result is True/False



OPERATOR	MEANING	EXAMPLE	RESULT
<	Less than	1<2	True
>	Greater than	1>2	False
<=	Less than or equal to	1<=2	True
>=	Greater than or equal to	1>=2	False
==	Equal to	1==2	False
!=	Not equal to	1!=2	True

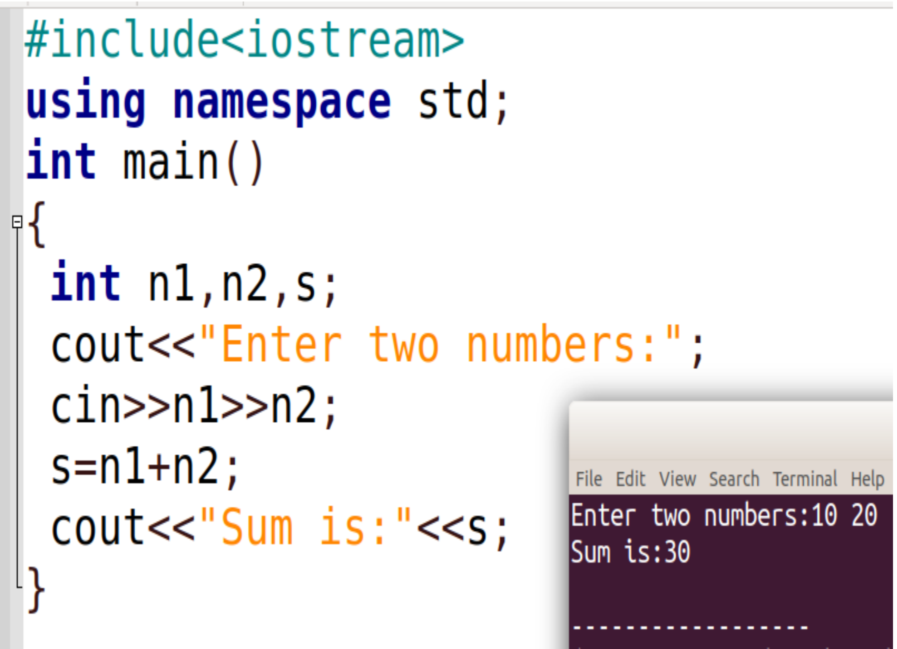
Logical Operators

Operator	Meaning	Example	Result
&&	Logical and	(5<2)&&(5>3)	False
||	Logical or	(5<2)||(5>3)	True
!	Logical not	!(5<2)	True

Input Output Operators

>> is the input operator. Also called get from or extraction operator.

<< is the output operator . Also called put to or insertion operator



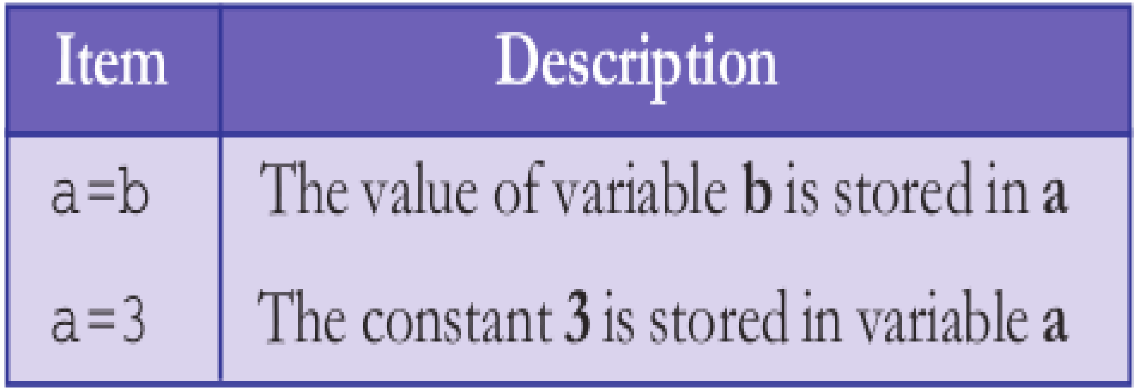
```
#include<iostream>
using namespace std;
int main()
{
 int n1,n2,s;
 cout<<"Enter two numbers:";
 cin>>n1>>n2;
 s=n1+n2;
 cout<<"Sum is:"<<s;
}
```

```
Enter two numbers:10 20
Sum is:30


-----------------
```

Assignment Operators

= is the assignment operator. Used to store a value in a variable.



Item	Description
a=b	The value of variable **b** is stored in **a**
a=3	The constant **3** is stored in variable **a**

Table 5.9 : Assignment operator

Expressions

Expressions are constituted by operators and operands

Arithmetic : Which uses arithmetic operators

Example :a+b,a-b

Relational : Which uses relational operators

Example:a<b, x==y,

Logical : Which uses logical operators

Example:a&&b , x>=y && x==20, !(a>b)

Statements in C++

Smallest executable unit of a programming language.

In C++ every statement is terminated by ;

There are four types of statements in C++

- **Declaration statements :** Used for declaring variables

```
int n1;
```

- **Input Statement :** Used for reading a value from keyboard

```
cin>>n1;
```

- **Assignment Statement:** Used for storing a value to a variable

```
s=n1+n2;
```

- **Output Statements:**Used for displaying a value to the screen.

```
cout<<s;
```

Declaring a Variable

Before using a variable in a program, it must be declared.

The general format for variable declaration is

DataType   VariableName1,Variable Name2....;

Example :

```
int n;
float n1,n2;
 char ch;
```

Cascading of I/O operators

The multiple use of input or output operators in a single statement is called cascading of I/O operators.

```
  cin>>n1;

  cin>>n2;

 cin>>n3;
```

Can be written as: `cin>>n1>>n2>>n3;`

Similarly

```
 cout<<”Sum is:”;

 cout<<s;
```

Can be written as `cout<<”Sum is:”<<s;`

Which among the following is an insertion operator

a) >>   b) <<   c) >   d) <

Ans) <<

<<- Insertion /Output /Put to operation

>>- Extraction/Input /Get from operation

Differentiate = and ==

Ans) = is assignment operator

== is relational operator

What are the outputs of below operations

i) 15%4 ii) 5/2 iii) 5<6 iv) !(4>2)

Ans) i) 3

ii) 2

Iii) True(Any non zero number)

iv) False/0

% gives reminder after division(modulus)

Output of relational /Logical operators are True/ False

AV
Vlog

SUBSCRIBE

To watch Video Class Click Here

Prepared by

Asees V

9495317820