UNIT-2

Fundamental of C++ Programming

- **HISTORY OF C++**:-

C++ is an object oriented programming language. It is an extended version of c language. It was developed by 'Byrne stroustrup' in 1980 at AT & T bell lab in USA.

Stroustrup is an ad mirror of an 'Simula'(programming language) and strong supporter of c programming. He wants to combined the base of both language and create a more powerful language that would support object orient6ed programming features and still retains the power of c. the result was C++. He initially called the language "c with classes".

However, in 1983 the name was changed C++ by 'rick mescitti' where ++ is c increment operators. C++ allows programmer to manage large and more complex program.

Since C++ have 3 major revision with adding to altering the language. The 1st revision was in 1985. And the 2nd in 1990. The 3rd occurs during the standardization of C++ in 1997.

ANSI and ISO standardization committee was form the 1st draft of proposed standard was created on January 25 in 1994. The final draft was passed out of committee on nov. 14 in 1997. C+ is superset of c. most of what we already known about c applies to C++ also. Therefore, almost all c programs are also C++ programs will present in c program to run under C++ compiler.

Structure of C++ Program:-

Include files
Class declaration
Class member function definition
Main function program

A typical C++ program contain a sections as shown in figure above. Those sections may be placed in separate code, files and then compiler independently and jointly. It is a common practice to organize a program into 3 separate files. The class declarations are placed in header file and the definition of class member function placed into another file. This approach enables the programmer to separate abstract specification or the interface from implementation details finally, the main program that uses the class are placed in third file which includes the previous two files as well as another file required.

C++ Tokens:-

The smallest individual unit of a program is known as 'Tokens'. C++ has the following tokens:-

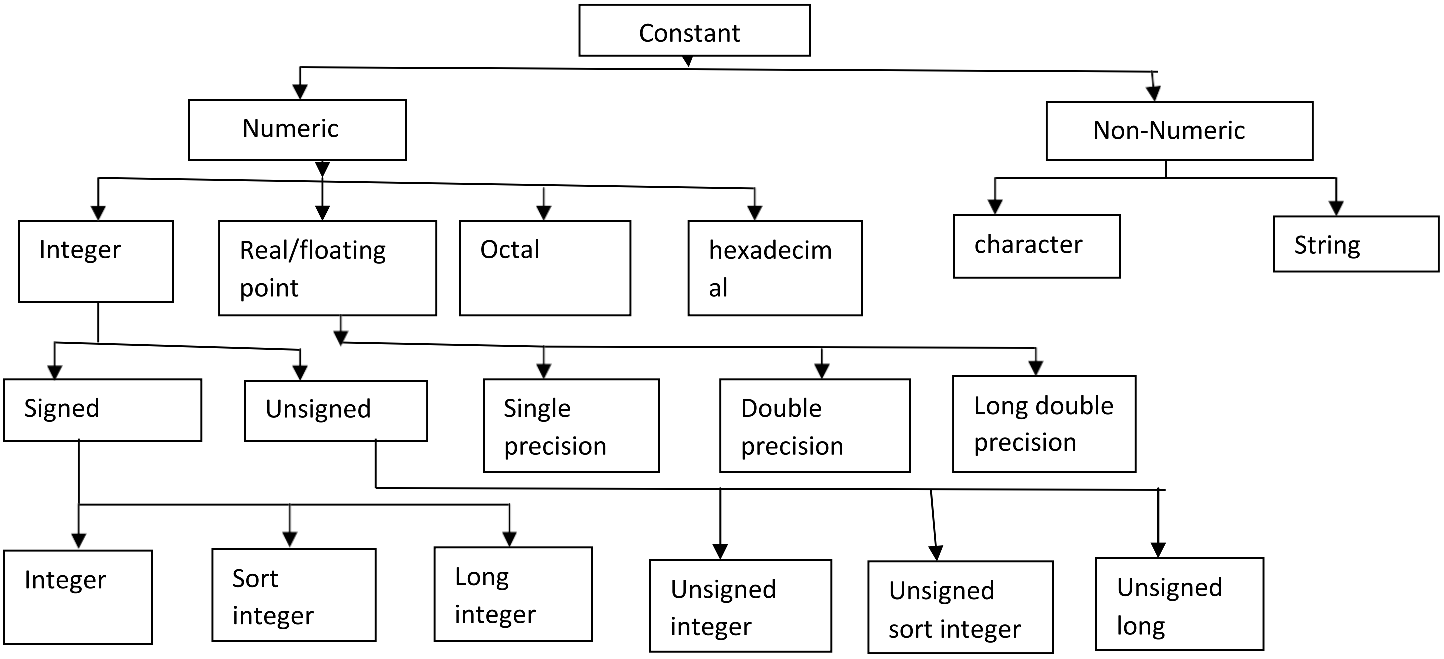
- Keyword
- Identifier
- Constant
- String

- Operators.

KEYWORD:-Keyword are reserved word and it can not be changed, it can not be used as a name for the program variable or other user defined program elements. A number of keywords can be added to ANSI-C. some of them are given below:-

- Catch
- Class
- New
- Delete
- Friend
- Inline
- Operator
- Private
- Public
- Protected
- Template
- This
- Try etc. Identifiers

& constant:-



- **Constant(Literals)**:-

  characters are fixed values whose values can not be changed during the execution of a program. It can be classified into two types:-

  i. Numeric constant
  ii. Non-numeric constant

i. **Numeric constant**:-

Numeric constants are positive or negative number. There are four types of numeric constant.

a. Integer constant
b. Floating point constant
c. Octal constant
d. Hexa-decimal constant

An **integer constant** do not contains decimal points. It can be signed orunsigned.

The signed integer constant contains both positive and negative values. Where as unsigned integer contains only positive values. An integer contains may either be sort integer or long integer.

Positive or negative numbers are represented in exponential form which can be represented by **floating point constant** consist of single precision (6 digit), double precision (10 digit) and long double precision (19 digit).

Hexa-decimal constant(hex constant):-

Hexa-decimal numbers are integer numbers of base 16 and their digit are 0 to 9, A to F(a to f). in C++, it is normally represented using character ‘x’. some examples of hexa-decimal constants are:- 0x2, 0x21, 22x35.

Octal constant:-

octal numbers are integer numbers of base 8 and their digits are 0 to 7. It is normally represented by a constant beginning with character zero(0). Ex:- 056, 011, 054 etc.

<u>Non-Numeric constant</u>

- Character constant:-

  Any character beginning to ASCII character set is considered as character constant. The character constant are enclosed in single quotation (‘ ’) mark.
- String constant:-

A string constant is a sequence of alphanumeric characters enclosed in double quotation mark (“ ”) whose maximum length is 255 characters.

Ex:-

i. “xyz”

ii. “The C++ variable”.

- **Data type**:-

Data types are used to identify the data and associated operation of handling it. Data can be many types such as:-

a. Integer
b. Float
c. Character etc.

C++ data types are classified in three categories.

a. Fundamental/Basic data type
b. Derived data type.
c. User defined data type.

a. Fundamental data type:-

Fundamental data types are those data type which is not the composition of other data types. There are five fundamental data types in C++. These are:-

i. integer (int)
ii. character (char)
iii. float
iv. double
v. void

other fundamental data types are :-

<u>**DATA TYPE**</u>	<u>**Size**</u>
char	1 byte
unsigned char	1byte
signed char	1 byte
int	2 byte
unsigned int	2 byte
short int	2 byte
unsigned short int	2byte
signed int	2 byte
long int	4 byte
signed long int	4 byte
unsigned long int	4 byte

float	4 byte
double	8 byte
long double	10 byte
signed int	2 byte

b. **Derived data type**:-

These are the data types which are derived from the fundamental data types.

Ex. Of derived data types are:-

i. Array
ii. Pointer.
iii. Reference etc.

i. **Array**:-

An array is a collection of identical data objects which are stored in consecutive memory location under a common heading or a variable name. in other word, an array is a group or a label of values referred to by the same variable name. the individual values in an array are called element. Array elements are also variables.

Array are variables value of the same type which have a single name followed by an index.

Syntax to declare an array:-

```
data type array_name[size];
```

```
Ex:- int arr[10];
```

here, arr can have maximum 11 elements and elements are identified by index value from to size of array as:-

```
arr[0]= 1;
arr[1]= 4;
 arr[2] = 7;
arr[3] = 9;
arr[4] = 5;
arr[5]= 10;
```

The chare type array i.e:- an array declared as char type is called ‘string’.

```
Char s[10];
s= “Maharaja”;
```

Q. WAP for the demonstration of array.

```
#include<iostream.h>
#include<conio.h> void
main()
{
  int a[100]; int
  i,n;
  cout<<"\n Enter size of array ="; cin
  >>n;
  cout<<"\n Enter the array elements="; for
 (i=0;i<n;i++)
 cin>>a[i];
 cout<<"\n Array elements will be \n"; for
 (i=0;i<n;i++)
cout<<"\n \t"<<a[i];
getch();
}
```

Pointer:-

A pointer is a variable that contain the address of another variable, pointer can be initialized and declared by the following statements.

```
Ex:-    int x, *p;
        x=10;
        p=&x;
```

C++ adds the concept of constant pointer and pointer to constant.

i. **Constant pointer:-**

In constant pointer we can't modify the address that the pointer is

initialize too. Ex:-

```
char * const p= "good";
      char s[4]= "Good";
```

ii. **Pointer to constant:-**

In pointer to constant pointer is declared that point to a constant. It points to any variable of correct type but the content of what a point to can't be change.

Ex:-

```
int const *p1= &m;
      where, m is a constant.
Int const x= 5;
```

- **Void pointer**:-

A void pointer uses a keyword ‘void’ for specifying the type of a pointer. Pointer define in this manner do not have any type associated with them and can hold any type of address of a variable.

The general format of void pointer is given below:-
Void * V-ptr;

Q. WAP to show the use of void pointer.

```
#include<iostream.h>
#include<conio.h> void
main()
 {
   int x=10; float
   y=2.5; void
   *ptr; ptr=&x;
   cout<<”\n Address of variable x=”<<ptr;
   ptr=&y;
   cout<<”\n Address of variable y=”<<ptr;
getch();
}
```

- **Reference variable**:-

Reference is an alternative name of an object and it is used to specify the argument and returns of function syntax:-

data type &reference_var_name = var_name; Ex:-
int x=10;
int &p= x;

c. **User defined data type**:-

C++ provides a feature, where we can define our own data type. That data type is known as user define data type. There are basically two types of user defined data types:-