variables and their types in C++

In this page, we will variable and their types in C++ which are following,

---

variable in C++

To store user-input in the program, we first declare the variable. The variable type, depending on the user input.

In a way, you can say that the variable is part of the memory location that stores the input given by the user.

Declaration of a variable in C++

The **variable-name** is an identifier that means any user-defined name,

as shown below. we declared some variable with different data-types. Which will be used to store different type of information.

```
data-type identifier;
```

Here identifier represents a **variable-name**,

Rule of a variable declaration in C++

Here we will go with `int` data type,

- we can declare an alphabet character and underscore, as a variable,

```
int x,x_,_x;
```

- a special type of character can't be a variable @, &, $, #, – ,) , (etc.

```
int #,(,@;  // wrong
```

- Because is possible to variable have long length size so,

```
char student_address[15],student_name[15],Name[15];
```

- uppercase and lowercase will be two different types of a variable such as a **name** and **Name** both will be different types of variable.

```
char Name[15],name[15];
```

- a reserved keyword which in C++ are already defined we can not use them as a variable.

```
int , float, class, char, void
```

- variables will never start with numbers,

```
int 2x ,3y;


char 1Name[15];
```

However, the following are possible,

```
int x2,y3;


char Name1[15];
```

suggestion: always use a meaningful name of a variable, such as

```
char x[15],y[15];
```

The Declared variable above does not indicate which type of data they will store while using a meaningful name of variables, such as,

```
char first_name[15], last_name[15];
```

now here we know variable **first_name** and **last_name** is used to store first and last name of a person so always give a meaningful name of a variable/identifier, This can make debugging the program a little easier.

C++ variable initialization

when a variable is declared, then at the same time assigning some value or assigning it, is called variable initialization. see example below

```
int age=25; or int a(25);  // for numeric type

char name = 'R';        // for single character

char name [6]= "Rahul";   // for sentence
```

`int` and `char` are data-type.

Rvalue – This is the name of a variable (e.g. **age, name**).
Lvalue – This is the value which is assigned to the variable (e.g. **25, Rahul**).

Example of a variable in C++

In the below Program two variables are declared **age** and **name,** where **name** variable used to store a person's name and variable **age** used to store a person's age. both inputs is given by the user.

```
#include<iostream.h>

using namespace std;



 int main()

  {

     int age;

     char name[6];



     cout<<"Enter age : ";

     cin>>age;
```

```
    cout<<"Enter Name: "
    cin>>name;
//displaying
    cout<<"Age : "<<age<<endl;
    cout<<"Name: "<<name;
    return 0;
  }
```

OUTPUT

```
Enter age : 25
Enter Name: Rahul
Age :25
Name:Rahul
```

variable Types in C++

C++ supports 3 types of variables which following are,

- local variable
- global variable
- reference variable

The declaration of these variables is the same as the normal variable but by changing the place of declaration, their scope also changes.

local variable in C++

Those variables that are declared within a function or block are called local variables.

These types of variable are visible or active within the function. That is when the function executes then local variables active, as the function ends, these variables are automatically destroyed. These can not be accessed from outside the function.

Here, **l** is a local variable which is defined inside the function of `int main()`

```
int main()

{

  int l=20;

  ..........



}
```

global variable in C++

Those variables that are declared outside a function or block are called Global variables.

They can be accessed from any part of the program or from the function.

Here, **g** is a global variable which is defined outside the function `int main()`

```
int g=10;
```

```
int main()

{

  ..........

 }
```

The program of local and global variables is given below-

Example of local and Global variable in C++

Here is the complete program,

```
#include<iostream.h>

using namespace std;

int g=10; // global variable declaration

int main()

 {

   int l=20; // local variable declaration

   cout<<"global variable: "<<g<<endl;

   cout<<"local variable: "<<l;

return 0;

 }
```

OUTPUT

```
global variable: 10
```

```
local variable: 20
```

In a large program that contains lots of function or blocks, instead of declaring multiple variables for each function, we can declare a single global variable.

another variable is found in C++, which is not found in C language.

reference variable in C++

The reference variable is an alternative to the already a declared variable, that is, two such variables whose names are different but their value is the same.

SYNTAX

```
data-type variable-name;

data-type &reference-variable = variable-name;
```

In this, the value is assigned in the first variable and in the second variable assigns the first variable. In this way, the first variable's value also assigning to the second variable automatically.

let's see with an Example

In below, **x** is an int type variable and **r** is a reference variable,

```
int x = 10;

int &r = x;
```

Here **r** is the alternative variable of **x**, where **x** represented by variable **r**

```
cout<<x; // 10

cout<<r; // 10
```

Example of reference variable in C++

```
#include<iostream.h>

using namespace std;

int main()

{

  int x=10;

  int &r = x; // reference variable

  cout<<"orginal variable: "<<x<<endl;

  cout<<"reference variable: "<<r;

  return 0;

 }
```

OUTPUT

```
Original variable: 10

reference variable: 10
```

Related Exercise

- C++ basic program
- Find out a variable size in own System in C++
- how to store full name in C++