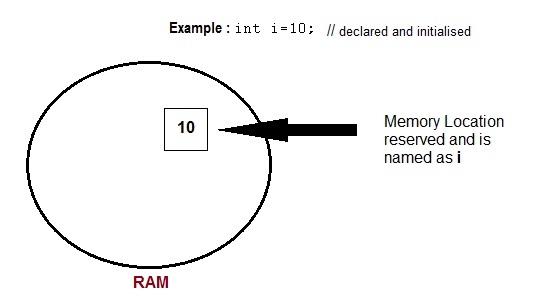
### What are Variables

Variable are used in C++, where we need storage for any value, which will change in program. Variable can be declared in multiple ways each with different memory requirements and functioning. Variable is the name of memory location allocated by the compiler depending upon the datatype of the variable.



#### Basic types of Variables

Each variable while declaration must be given a datatype, on which the memory assigned to the variable depends. Following are the basic types of variables,

|  |  |
| --- | --- |
| bool | For variable to store boolean values( True or False ) |
| char | For variables to store character types. |
| int | for variable with integral values |
| float and double are also types for variables with large and floating point values | |

#### Declaration and Initialization

Variable must be declared before they are used. Usually it is preferred to declare them at the starting of the program, but in C++ they can be declared in the middle of program too, but must be done before using them.

*Example* :

int i; // declared but not initialised

char c;

int i, j, k; // Multiple declaration

Initialization means assigning value to an already declared variable,

int i; // declaration

i = 10; // initialization

Initialization and declaration can be done in one single step also,

int i=10; //initialization and declaration in same step

int i=10, j=11;

If a variable is declared and not initialized by default it will hold a garbage value. Also, if a variable is once declared and if try to declare it again, we will get a compile time error.

int i,j;

i=10;

j=20;

int j=i+j; //compile time error, cannot redeclare a variable in same scope

#### Scope of Variables

All the variables have their area of functioning, and out of that boundary they don't hold their value, this boundary is called scope of the variable. For most of the cases its between the curly braces,in which variable is declared that a variable exists, not outside it. We will study the storage classes later, but as of now, we can broadly divide variables into two main types,

* Global Variables
* Local variables

#### Global variables

Global variables are those, which ar once declared and can be used throughout the lifetime of the program by any class or any function. They must be declared outside the main() function. If only declared, they can be assigned different values at different time in program lifetime. But even if they are declared and initialized at the same time outside the main() function, then also they can be assigned any value at any point in the program.

*Example* : Only declared, not initialized

include <iostream>

using namespace std;

int x; // Global variable declared

int main()

{

x=10; // Initialized once

cout <<"first value of x = "<< x;

x=20; // Initialized again

cout <<"Initialized again with value = "<< x;

}

#### Local Variables

Local variables are the variables which exist only between the curly braces, in which its declared. Outside that they are unavailable and leads to compile time error.

*Example*:

include <iostream>

using namespace std;

int main()

{

int i=10;

if(i<20) // if condition scope starts

{

int n=100; // Local variable declared and initialized

} // if condition scope ends

cout << n; // Compile time error, n not available here

}

#### Some special types of variable

There are also some special keywords, to impart unique characteristics to the variables in the program. Following two are mostly used, we will discuss them in details later.

1. **Final** - Once initialized, its value cant be changed.
2. **Static** - These variables holds their value between function calls.

*Example* :

include <iostream>

using namespace std;

int main()

{

final int i=10;

static int y=20;

{