SNS COLLEGE OF TECHNOLOGY

Coimbatore – 35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

23ITT101 – PROGRAMMING IN C & DATA STRUCTURES
I YEAR - II SEM

UNIT I – INTRODUCTION TO C

VARIABLES

C VARIABLES

-

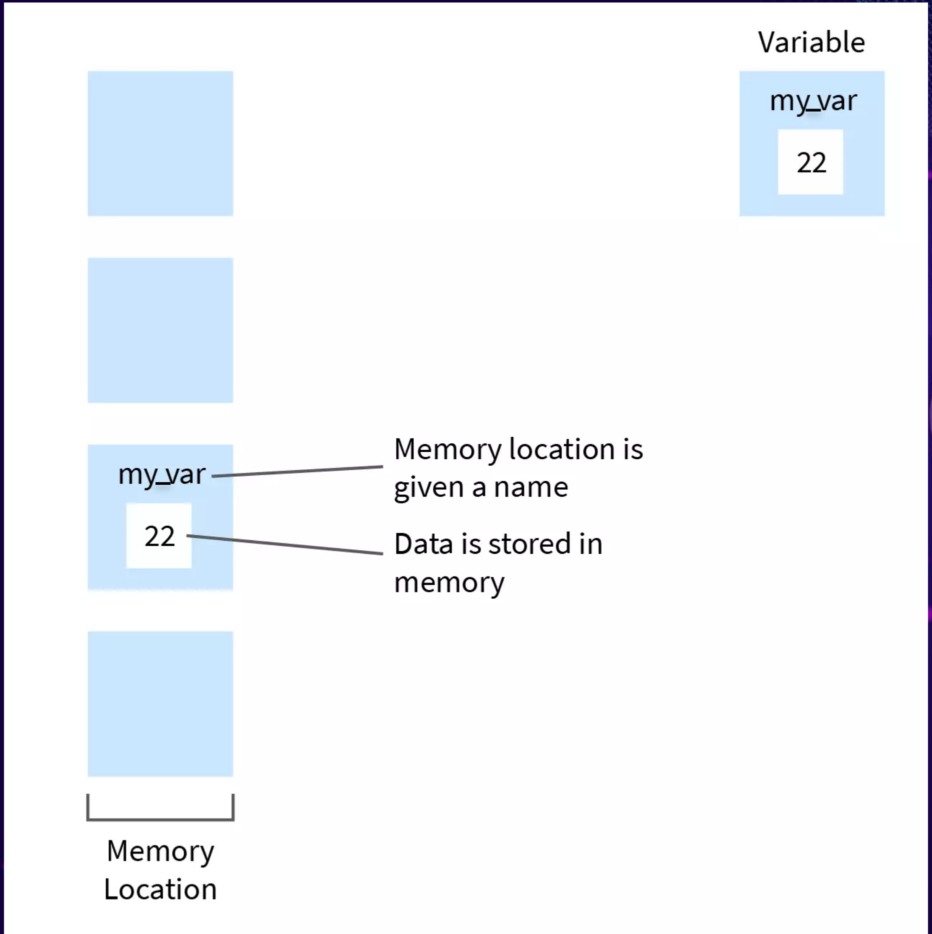
DECLARATION & INITIALIZATION

Variables

- ✍ Variable is basically nothing but the name of a memory location that we use for storing data.
- ✍ Variables are the **storage areas** in a code that the program can easily manipulate.
- ✍ Every variable in C language has some specific type- that determines the layout and the size of the memory of the variable, the range of values that the memory can hold, and the set of operations that one can perform on that variable.
- ✍ The name of a variable can be a composition of digits, letters, and also underscore characters. The name of the character must begin with either an underscore or a letter.
- ✍ In the case of C, the lowercase and uppercase letters are distinct. It is because C is case-sensitive in nature.

Example:

```
int my_var = 22;
```



Rules for naming a variable

1. The name of the variable must not begin with a digit.

2. A variable name can consist of digits, alphabets, and even special symbols such as an underscore (_).

3. A variable name must not have any keywords, for instance, float, int, etc.

4. There must be no spaces or blanks in the variable name.

5. The C language treats lowercase and uppercase very differently, as it is case sensitive. Usually, we keep the name of the variable in the lower case.

Examples

```
int first_name;      // it is correct
int var1;            // it is correct
int 1var;            // it is incorrect – the name of the variable should not start using a number
int my$var;          // it is incorrect – no special characters should be in the name of the variable
char else;           // there must be no keywords in the name of the variable
int my var;          // it is incorrect – there must be no spaces in the name of the variable
```

Declaration of Variables

- ✓ Declaration of a variable in a computer programming language is a statement used to specify the variable name and its data type.
- ✓ Declaration tells the compiler about the existence of an entity in the program and its location.

Syntax

```
data_type variable_name1, variable_name2, variable_name3;

OR

data_type variable_name;
```

Example: `int radius;` `char name[50],class;` `float kilometer;`

Initialization of Variables

Initialization is the process of assigning a value to the Variable. Every programming language has its own method of initializing the variable.

If the value is not assigned to the Variable, then the process is only called a Declaration

Syntax

```
data_type variable_name1=value, variable_name2=value,
                variable_name=value;

                          OR

          data_type variable_name=value;
```

Example: `int a=10, b=17, c=8;` `char s=‘a’;`

Types of Initialization

There are two types of variable initialization and they are,

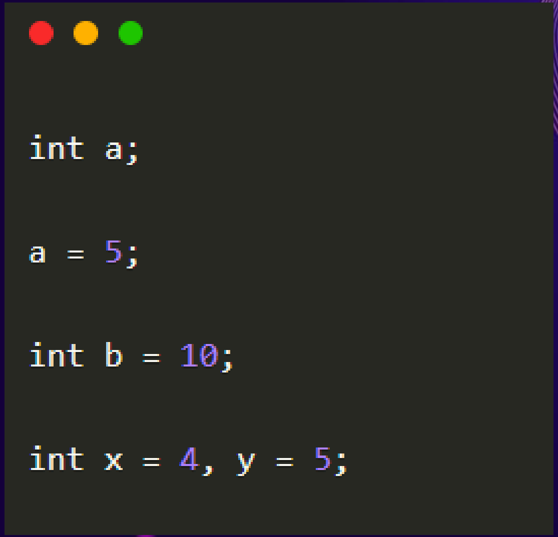
- **Static Initialization**
- **Dynamic Initialization**

Static Initialization

In this method, the variable is assigned a value in advance.

Here, the values are assigned in the declaration statement.

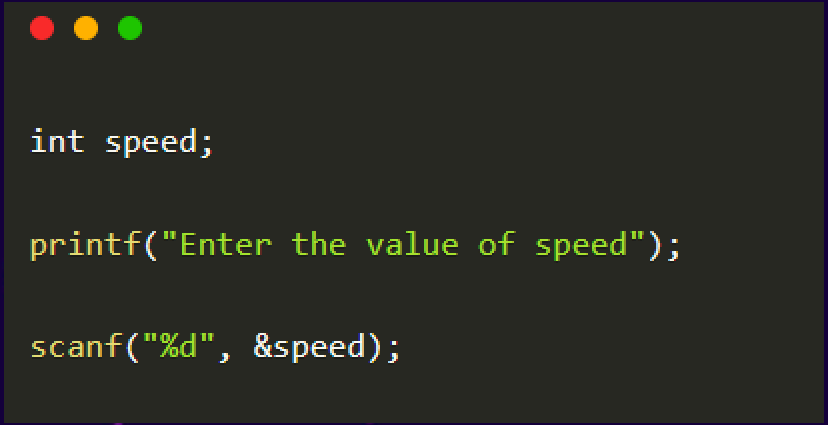
Static Initialization is also known as **Explicit Initialization.**



```
int a;

a = 5;

int b = 10;

int x = 4, y = 5;
```

Dynamic Initialization

In this method, the variable is assigned a value at the run-time. The value is either assigned by the function in the program or by the user at the time of running the program. The value of these variables can be altered every time the program runs. Dynamic Initialization is also known as **Implicit Initialization**.



```
int speed;

printf("Enter the value of speed");

scanf("%d", &speed);
```

Rules to declare and Initialize variables

1. Variable names must begin with a letter, underscore, non-number character. Each language has its own conventions.
2. Always use the ‘=’ sign to initialize a value to the Variable.
3. Do not use a comma with numbers. Eg: int a = 5,6,7;
4. Once a data type is defined for the variable, then only that type of data can be stored in it. For example, if a variable is declared as Int, then it can only store integer values.
5. A variable name once defined can only be used once in the program. You cannot define it again to store another type of value.
6. If another value is assigned to the variable which already has a value assigned to it before, then the previous value will be overwritten by the new value.

Scope of Variables

A scope in any programming is a region of the program where a defined variable can have its existence and beyond that variable cannot be accessed.

There are three places where variables can be declared in C programming language:

1. **Inside a function or a block is called local variable,**

2. **Outside of all functions is called global variable.**

3. **In the definition of function parameters which is called formal parameters.**

Local Variables

Variables that are declared inside a function or block are called local variables. They can be used only by statements that are inside that function. Local variables are not known to functions outside their own.

Example

```
#include <stdio.h>
main ()
{
/* local variable declaration */
int a, b, c;
/* actual initialization */
a = 10;
b = 20;
c = a + b;
printf ("value of a = %d, b = %d and c = %d \n", a, b, c);
}
```

Global Variables

Global variables are defined outside of a function, usually on top of the program. The global variables will hold their value throughout the lifetime of your program and they can be accessed inside any of the functions defined for the program. A global variable can be accessed by any function. That is, a global variable is available for use throughout your entire program after its declaration.

Example

```
#include <stdio.h>
/* global variable declaration */
int g;
main ()
{
/* local variable declaration */
int a, b;
a = 10;
b = 20;
g = a + b;
printf ("value of a = %d, b = %d and g = %d\n", a, b, g);
}
```

FAQ

1. Is the following statement a declaration or definition? ~ const int i;

a. Definition
b. Declaration
c. Error
d. None

2. Which declaration is correct?

a. int length;
b. float double;
c. float long;
d. long int;

3. Which statement is correct for a chained assignment?

a. int x = 5;
b. int x, y = 10;
c. int x = y = 10;
d. Both B and C