Variables in C

A **variable** is a name of the memory location. It is used to store data. Its value can be changed, and it can be reused many times. float a=5.12;

It is a way to represent memory location through symbol so that it can be easily identified.

Let's see the syntax to declare a variable:

1. type variable_list;

The example of declaring the variable is given below:

```
1. int a;
2. float b;
3. char c;
```

Here, a, b, c are variables. The int, float, char are the data types.

We can also provide values while declaring the variables as given below:

```
1. int a=10,b=20;//declaring 2 variable of integer type
2. float f=20.8;
3. char sizan='A';
```

Rules for defining variables

- A variable can have alphabets, digits, and underscore.
- A variable name can start with the alphabet, and underscore only. It can't start with a digit.
- No whitespace is allowed within the variable name.
- A variable name must not be any reserved word or keyword, e.g. int, float, etc.

Types of Variables in C

There are many types of variables in c:

1. local variable
2. global variable

3. static variable
4. automatic variable
5. external variable

Local Variable

A variable that is declared inside the function or block is called a local variable.

It must be declared at the start of the block.

```
1. int main(){
2. int x=10;//local variable
3. }
```

You must have to initialize the local variable before it is used.

Global Variable

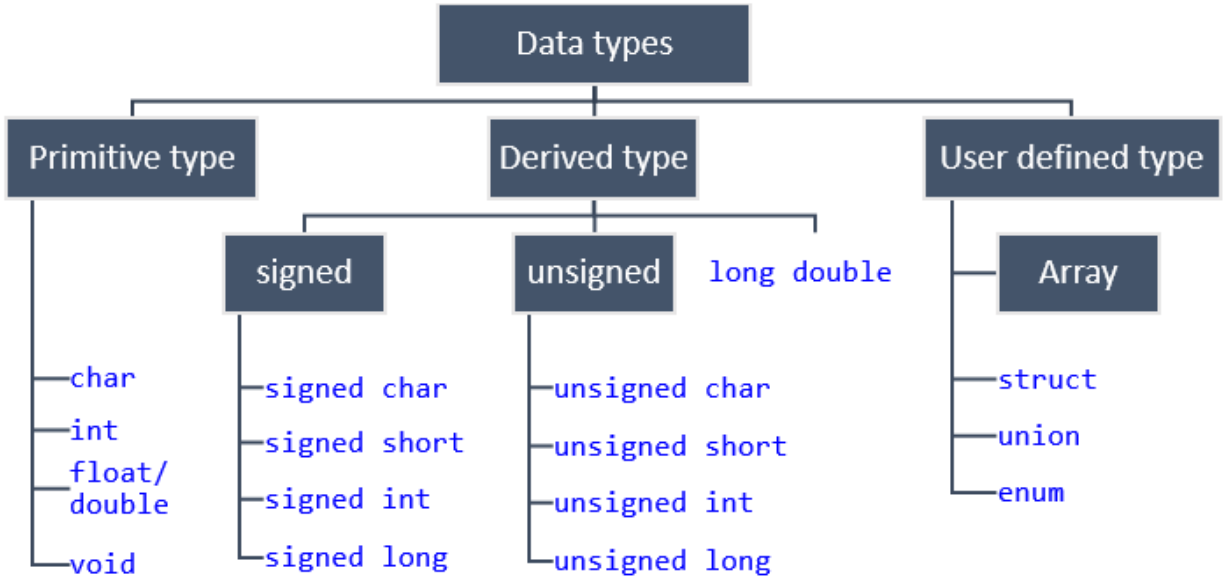
A variable that is declared outside the function or block is called a global variable. Any function can change the value of the global variable. It is available to all the functions.

It must be declared at the start of the block.

```
1.
2. int value=20;//global variable
3. int main(){
4. int x=10;//local variable
5. float b=4.5; // local
6. }
```

Data Types in C

A data type specifies the type of data that a variable can store such as integer, floating, character, etc. float a=5.5



Type	Size (bits)	Size (bytes)	Range
char	8	1	-128 to 127
unsigned char	8	1	0 to 255
int	16	2	-2^{15} to $2^{15}-1$
unsigned int	16	2	0 to $2^{16}-1$
short int	8	1	-128 to 127
unsigned short int	8	1	0 to 255
long int	32	4	-2^{31} to $2^{31}-1$
unsigned long int	32	4	0 to $2^{32}-1$
float	32	4	3.4E-38 to 3.4E+38
double	64	8	1.7E-308 to 1.7E+308
long double	80	10	3.4E-4932 to 1.1E+4932

Difference between primitive and non primitive data structures

Primitive Data Structure	Non-Primitive Data Structure
Primitive Data Structure are predefined in the language	Non-Primitive Data structure are not defined in language and created by the programmer
Primitive Data structures will have a certain value	Non Primitive Data structure can have NULL value
The size depends upon the type of data structure	The size of non primitive data structure are not fixed ar[1000]
The primitive data structure starts with lowercase	The non primitive data type starts with an uppercase
Can be used to call methods to perform operations	Cannot be used.

Primitive Data Type

Primitive data structures are the predefined types of data that are supported in the programming language.

These are the fundamental data types of the language. For example float, integer, character, and pointer.

Integer – The integers are used to represent the numeric data. The integer generally stores whole numbers which can be positive and negative. Ex: 1, 2, 300, -55

Float – The float data type is used to represent the fractional numbers or numbers with decimal figures in the languages. Ex: 2.65, 5.625

Boolean – The Boolean data type can only take up to two values that are TRUE or FALSE. Mostly, the boolean values are used for conditional testing.

if(5%2==1) then true

otherwise false

Character – The character data type is used to store single word characters both upper and lower case such as 'Z' or 'z'. 'a','b'.

Non-Primitive Data Structures

The Non-Primitive Data Structures are created with the help of the primitive data structures. The Non-primitive data structures are more complicated than primitive data structures but highly useful.

The non-primitive data types are the types that are defined by the programmer. They are further classified into **linear** and **nonlinear** data types.

The **linear** data types are storing the data inside the memory in a sequence one after.

When you have to retrieve the data from the linear data structure then you have to just start from one place and you can find other data in a sequence.

The various types of Linear data structures are –

Array – An array data structure can hold a fixed number of primitive data structures (such as integer, character, etc.). The basic operation that can be performed on the array is insertion, deletion, searching, updating, and traversing an array.

The main difference between primitive and non-primitive data types are:

- **Primitive types** are predefined (already defined) in C. ...
- **Non-primitive types** can be used to call methods to perform certain operations, while **primitive types** cannot.