Basic Data Types and Variables in c

Variables in C:

Variable is basically nothing but the name of a memory location that we use for storing data. We can change the value of a variable in C or any other language, and we can also reuse it multiple times. We use symbols in variables for representing the memory location- so that it becomes easily identifiable by any user.

Use of the Variables in C

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. Let us look at some more ways in which we name a variable.

Rules for Naming a Variable in C

We give a variable a meaningful name when we create it. Here are the rules that we must follow when naming it:

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.

Let us look at some of the examples,

---

int var1; // it is correct

int 1var; // it is incorrect – the name of the variable should not start using a number

int my_var1; // it is correct

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

int COUNT; // it is a new variable

int Count; // it is a new variable

int count; // it is a valid variable name

Data Type of the Variable

We must assign a data type to all the variables that are present in the C language. These define the type of data that we can store in any variable. If we do not provide the variable with a data type, the C compiler will ultimately generate a syntax error or a compile-time error.

The data Types present in the C language are float, int, double, char, long int, short int, etc., along with other modifiers.

Types of Primary/ Primitive Data Types in C Language

The variables can be of the following basic types, based on the name and the type of the variable:

Type of Variable	Name	Description	Uses
char	Character	It is a type of integer. It is typically one byte (single octet).	We use them in the form of single alphabets, such as X, r, B, f, k, etc., or for the ASCII character sets.
int	Integer	It is the most natural size of an integer used in the machine.	We use this for storing the whole numbers, such as 4, 300, 8000, etc.
float	Floating-Point	It is a floating-point value that is single precision.	We use these for indicating the real number values or decimal points, such as 20.8, 18.56, etc.
double	Double	It is a floating-point value that is double precision.	These are very large-sized numeric values that aren't really allowed in any data type that is a floating-point or an integer.
void	Void	It represents that there is an absence of type.	We use it to represent the absence of value. Thus, the use of this data type is to define various functions.

Let us look at a few examples,

```
// int type variable in C
int marks = 45;
// char type variable in C
char status = 'G';
// double type variable in C
double long = 28.338612;
// float type variable in C
float percentage = 82.5;
char c, ch;
```

int p, q, r;

double d;

float f, salary;

Classification of Variables in C

The variables can be of the following basic types, based on the name and the type of the variable:

- **Global Variable**: A variable that gets declared outside a block or a function is known as a global variable. Any function in a program is capable of changing the value of a global variable. It means that the global variable will be available to all the functions in the code. Because the global variable in c is available to all the functions, we have to declare it at the beginning of a block. Explore, Global Variable in C to know more.
  Example,
  int value=30; // a global variable
  void function1(){
  int a=20; // a local variable
  }
- **Local Variable:** A local variable is a type of variable that we declare inside a block or a function, unlike the global variable. Thus, we also have to declare a local variable in c at the beginning of a given block.
  Example,
  void function1(){
  int x=10; // a local variable
  }
  A user also has to initialize this local variable in a code before we use it in the program.
- **Automatic Variable:** Every variable that gets declared inside a block (in the C language) is by default automatic in nature. One can declare any given automatic variable explicitly using the

keyword *auto*.
Example,
void main(){
int a=80; // a local variable (it is also automatic variable)
auto int b=50; // an automatic variable
}

- **Static Variable:** The static variable in c is a variable that a user declares using the *static* keyword. This variable retains the given value between various function calls.
Example, void function1(){

int a=10; // A local variable

static int b=10; // A static variable

a=a+1;

b=b+1;

printf("%d,%d",a,b);

}

If we call this given function multiple times, then the local variable will print this very same value for every function call. For example, 11, 11, 11, and so on after this. The static variable, on the other hand, will print the value that is incremented in each and every function call. For example, 11, 12, 13, and so on.

- **External Variable:** A user will be capable of sharing a variable in multiple numbers of source files in C if they use an external variable. If we want to declare an external variable, then we need to use the keyword *extern*.
Syntax, extern int a=10;// external variable (also a global variable)