C Variables, Data Types

Last update on April 14 2018 06:14:55 (UTC/GMT +8 hours)

What is variable?

Variables in C have the same meaning as variables in algebra. A variable in C is a storage unit, which sets a space in memory to hold a value and can take different values at different times during program execution.

Rules to construct a valid variable name

1. A variable name may consists of letters, digits and the underscore (_) characters.

2. A variable name must begin with a letter. Some system allows to starts the variable name with an underscore as the first character.

3. ANSI standard recognizes a length of 31 characters for a variable name. However, the length should not be normally more than any combination of eight alphabets, digits, and underscores.

4. Uppercase and lowercase are significant. That is the variable **Totamt** is not the same as **totamt** and **TOTAMT.**

5. The variable name may not be a C reserved word (keyword).

Some valid variable names

Total	Amount	ctr	name1
n1	M_age	AMOUNT	

Some invalid variable names

13th	(name)	111	%nm

Naming Conventions

Generally, C programmers maintain the following conventions for naming variables.

- Start a variable name with lowercase letters.
- Try to use meaningful identifiers
- Separate "words" within identifiers with mixed upper and lowercase (for example empCode) or underscores (for example emp_code).
- For symbolic constants use all uppercase letters (for example #define LENGTH 100, #define MRP 45).

Keywords and Identifiers

Every C word is classified as either a keyword or an identifier. Each keyword has a specific meaning and these meanings cannot be changed. Keywords serve as basic building blocks for program statements. There are only 32 keywords in C. The list of all keywords in ANSI C is listed in the following table. All keywords must be written in lowercase.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Constants

Constants in C refer to a specific quantity that doesn't change during the execution of a program.

Types of Constants

- Integer Constants
- Real Constants
- Single Character Constants
- String Constants

Rules to construct Integer Constant

- An integer constant refers to a sequence of digits. The three types of integers are decimal, octal and hexadecimal
- No embedded blanks, commas, and non-numeric character are permitted within an integer constant.
- An integer constant must contain one digit
- Decimal integers consist set of digits, 0 to 9 without any decimal point and can be preceded by an optional +ve or -ve sign.
- An octal integer constant contains any combination of digits between 0 and 7 with a leading 0.
- A hexadecimal constant contains any combination of digits between 0 and 9 and can also hold alphabets between A and F or a and f with prefix 0x or 0X. Alphabets A or a represents number 10 and F or f represents 15.
- The largest integer value for the 16-bit machine is 32767 and 2147483647 for a 32-bit machine.

Example of various valid numeric constants

Constant	Type	Constant	Type
241	Decimal Integer	047	Octal Integer
-973	Decimal Integer	053	Octal Integer
0	Decimal Integer	0X59	Hexadecimal Int
+4177	Decimal Integer	0x47F	Hexadecimal Int

Example of some invalid numeric constants

Invalid Constant
05 241
7,412
$120

Rules to construct Real Constant

- A real constant is a number that may have a fractional part.
- It could be either +ve or -ve.
- No embedded blanks, commas, and non-numeric character are permitted within a real constant.
- A real number may also be expressed in exponential notation. An exponent is an integer number with an optional plus or minus sign. Exponential is useful for representing the number which is very large or very small in magnitude.
-

Example of various valid real constants

0.0045	-.71
+45.203	0.45e3
-0.547	0.78e-4
337.	2.79E4
.478.	-4.69E-4

Data Types

Each program needs a certain kind of data for displaying a meaningful result. This certain kind of data are known as a data type.

ANSI C supports four classes of data types :

- Primary data types

- User-defined data types
- Derived data types
- Empty data set

All C compilers support four fundamental data types

Type	Range of values	I
char (Characters)	-128 to 127	a
int Integers (whole numbers)	-32768 to 32767	a
float Floating point (real numbers)	3.4e-38 to 3.4e+38	s
double (Double)	1.7e-308 to 1.7e+308	(

Character Types :

A single character can be defined as a character type data. Characters are usually stored in 8 bits of internal storage. The qualifier signed or unsigned may be explicitly applied to char. While unsigned chars have values between 0 and 255, signed chars have values from -128 to 127.

Integer Types :

C has three classes of integer storage, namely short int, int, and long int, in both signed and unsigned forms.

The keywords signed and unsigned are the two sign qualifiers which specify whether a variable can store positive or negative or both numbers.

The keyword signed uses one bit for a sign and 15 bits for the magnitude of the number in a 16-bit machine.

The keyword unsigned uses to store all the bits for the magnitude of the number and always positive.

Floating Point Types :

Floating point numbers are stored in 32 bits with 6 digits of precision. Floating point numbers are defined in C by the keyword float. When the accuracy provided by a float number is not sufficient, the type double can be used to define the number.

Double Point Types :

A double data type number uses 64 bits giving a precision of 14 digits. These are known as double precision numbers. Remember that double type represents the same data type that float represents but with a greater precision. To extend the precision further, we may use long double which uses 80 bits.

The following table shows the size and range of the type-specifiers on most common implementations :

Type	Size(bits)	Range
char or signed char	8	-128 to
unsigned char	8	0 to 25
int or signed int	16	-32768 to
unsigned int	16	0 to 655
short int or signed short int	8	-128 to
unsigned short int	8	0 to 25
long int or signed long int	32	-2147483648 to
unsigned long int	32	0 to 42949
float	32	3.4E-38 TO
double	64	1.7E-308 TO
long double	80	3.4E-4932 TO

Initialization of Variables

Variables are given initial values, or initialized, when declared. See the following examples :

char abc = 'X';
int marks =77;

```
float amount = 45.23;
```

Write initialized variables on a separate line and a comment beside the variable name is a good idea. See the following examples :

```
int qty; /* quantity of an item */
float value = 12.10; /* purchase value of an item */
int marks; /* marks of a subject */
```