Character set

A character denotes any alphabet, digit or special symbol used to represent information. Valid alphabets, numbers and special symbols allowed in C are

Alphabets	A, B, ….., Y, Z
	a, b, ……, y, z
Digits	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Special symbols	~ ' ! @ # % ^ & * ( ) _ - + =	\ { }
	[ ] : ; " ' < > , . ? /

The alphabets, numbers and special symbols when properly combined form constants, variables and keywords.

Identifiers

Identifiers are user defined word used to name of entities like variables, arrays,functions, structures etc. Rules for naming identifiers are:

1) name should only consists of alphabets (both upper and lower case), digits and underscore (_) sign.

2) first characters should be alphabet or underscore

3) name should not be a keyword

4) since C is a case sensitive, the upper case and lower case considered differently, for example code, Code, CODE etc. are different identifiers.

5) identifiers are generally given in some meaningful name such as value, net_salary, age, data etc. An identifier name may be long, some implementation recognizes only first eight characters, most recognize 31 characters. ANSI standard compiler recognize 31 characters. Some invalid identifiers are 5cb, int, res#, avg no etc.

Keyword

There are certain words reserved for doing specific task, these words are known as **reserved word** or **keywords.** These words are predefined and always written in lower case or small letter. These keywords cann't be used as a variable name as it assigned with fixed meaning. Some examples are **int, short, signed, unsigned, default, volatile, float, long, double, break, continue, typedef, static,**

do, for, union, return, while, do, extern, register, enum, case, goto, struct, char, auto, const etc.

Data types
Data types refer to an extensive system used for declaring variables or functions of different types before its use. The type of a variable determines how much space it occupies in storage and how the bit pattern stored is interpreted. The value of a variable can be changed any time.

C has the following 4 types of data types

basic built-in data types: int, float, double, char

Enumeration data type: enum

Derived data type: pointer, array, structure, union

Void data type: void

A variable declared to be of type int can be used to contain integral values only—that is, values that do not contain decimal places. A variable declared to be of type float can be used for storing floating- point numbers (values containing decimal places). The double type is the same as type float, only with roughly twice the precision. The char data type can be used to store a single character, such as the letter *a*, the digit character *6*, or a semicolon similarly A variable declared char can only store character type value.

There are two types of type qualifier in c

Size qualifier: short, long

Sign qualifier: signed, unsigned
When the qualifier unsigned is used the number is always positive, and when signed is used number may be positive or negative. If the sign qualifier is not mentioned, then by default sign qualifier is assumed. The range of values for signed data types is less than that of unsigned data type. Because in signed type, the left most bit is used to represent sign, while in unsigned type this bit is also used to represent the value. The size and range of the different data types on a 16bit machine is given below:

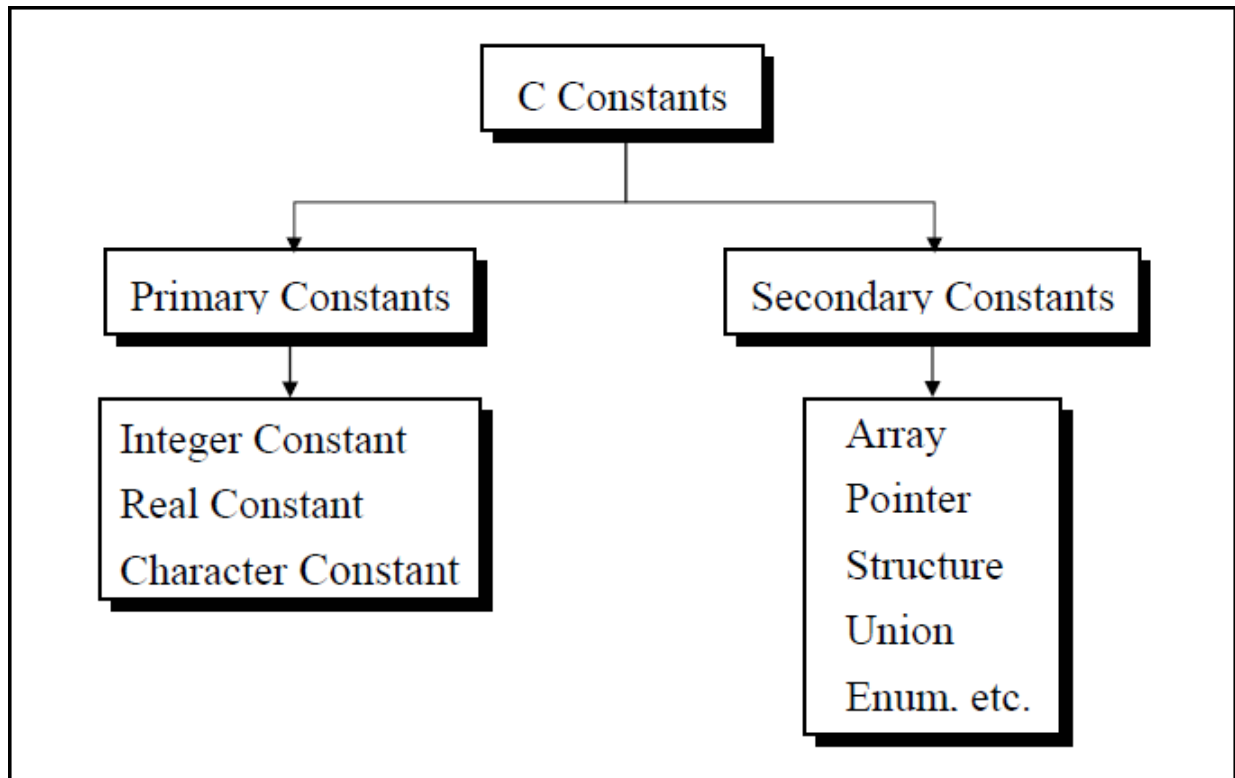| Basic data type	Data type with type qualifier	Size (byte)	Range
char	char or signed char	1	-128 to 127
	Unsigned char	1	0 to 255
int	int or signed int	2	-32768 to 32767
	unsigned int	2	0 to 65535
	short int or signed short int	1	-128 to 127
	unsigned short int	1	0 to 255
	long int or signed long int	4	-2147483648 to 2147483647
	unsigned long int	4	0 to 4294967295
float	float	4	-3.4E-38 to 3.4E+38
double	double	8	1.7E-308 to 1.7E+308
	Long double	10	3.4E-4932 to 1.1E+4932

Constants

Constant is a any value that cannot be changed during program execution. In C, any number, single character, or character string is known as a *constant*. A constantis an entity that doesn't change whereas a variable is an entity that may change.For example, the number 50 represents a constant integer value. The character string "Programming in C is fun.\n" is an example of a constant character string. C constants can be divided into two major categories:
Primary   Constants
Secondary Constants

These constants are further categorized as

```
                        ┌─────────────┐
                        │ C Constants │
                        └──────┬──────┘
              ┌────────────────┴────────────────┐
    ┌───────────────────┐            ┌─────────────────────┐
    │ Primary Constants │            │ Secondary Constants │
    └─────────┬─────────┘            └──────────┬──────────┘
    ┌─────────┴──────────┐           ┌──────────┴──────────┐
    │ Integer Constant   │           │ Array               │
    │ Real Constant      │           │ Pointer             │
    │ Character Constant │           │ Structure           │
    └────────────────────┘           │ Union               │
                                     │ Enum. etc.          │
                                     └─────────────────────┘
```

Numeric constant: Numeric constant consists of digits. It required minimum size of 2 bytes and max 4 bytes. It may be positive or negative but by default sign is always positive. No comma or space is allowed within the numeric constant and it must have at least 1 digit.

It is categorized a **integer constant** and **real constant.** An integer constants are whole number which have no decimal point. Types of integer constants are:

    Decimal constant: 0 ------9(base 10)
    Octal constant: 0 ------ 7(base 8)
    Hexa decimal constant: 0----9, A----- F(base 16)

   In decimal constant first digit should not be zero unlike octal constant first digit must be zero(as 076, 0127) and in hexadecimal constant first two digit should be 0x/ 0X (such as 0x24, 0x87A).

Real constant is also called floating point constant. To construct real constant we must follow the rule of ,

 -real constant must have at least one digit.
 -It must have a decimal point.
  -It could be either positive or negative.
 -Default sign is positive.
  -No commas or blanks are allowed within a real constant. Ex.: +325.34
    426.0
    -32.76

To express small/large real constant exponent(scientific) form is used where number is written in mantissa and exponent form separated by e/E. Exponent can be positive or negative integer but mantissa can be real/integer type, for example $3.6*10^5=3.6e+5$. By default type of floating point constant is double, it can also be explicitly defined it by suffix of f/F.

Character constant

Character constant represented as a single character enclosed within a single quote. These can be single digit, single special symbol or white spaces such as '9','c','$', ' ' etc. Every character constant has a unique integer like value in machine's character code as if machine using ASCII (American standard code forinformation interchange). Some numeric value associated with each upper and lower case alphabets and decimal integers are as:

        A ------------Z ASCII value (65-90)
         a ------------ z ASCII value (97-122)
        0-------------9   ASCII value (48-59)
         ;   ASCII value (59)

String constant

Set of characters are called string and when sequence of characters are enclosed within a double quote (it may be combination of all kind of symbols) is a string constant. String constant has zero, one or more than one character and at the end of the string null character(\0) is automatically placed by compiler. Some examples are ",sarathina" , "908", "3"," ", "A" etc. In C although same characters are enclosed within single and double quotes it represents different meaning such as "A" and 'A' are different because first one is string attached with null character at the end but second one is character constant with its corresponding ASCII value is 65.

Symbolic constant

Symbolic constant is a name that substitute for a sequence of characters and, characters may be numeric, character or string constant. These constant are generally defined at the beginning of the program as

#define name value , here name generally written in upper case for example

#define MAX 10

#define CH 'b'

#define NAME "sony"

Variables

Variable is a data name which is used to store some data value or symbolic names for storing program
computations and results. The value of the variable can be change during the execution. The rule for naming the variables is same as the naming identifier. Before used in the program it must be declared. Declaration of variables specify itsname, data types and range of the value that variables can store depends upon itsdata types.

Syntax

int a;

Variable initialization

When we assign any initial value to variable during the declaration, is called initialization of variables. When variable is declared but contain undefined value

then it is called garbage value. The variable is initialized with the assignment operator such as

Data type variable name=constant;

Example: int a=20;

Or int a;
a=20;

Expressions:

An expression is a combination of variables, constants, operators and function call. It can be arithmetic, logical and relational for example:-

int z= x+y   // arithmatic expression

a>b   //relational

a==b   //   logical
func(a, b)   // function call

Expressions consisting entirely of constant values are called *constant expressions*. So, the expression

121 + 17 - 110

is a constant expression because each of the terms of the expression is a constant value. But if i were declared to be an integer variable, the expression

$180 + 2 - j$

would not represent a constant expression.

Expressions