C Character Set

As every language contains a set of characters used to construct words, statements, etc., C language also has a set of characters which include **alphabets, digits**, and **special symbols**. C language supports a total of 256 characters.

Every C program contains statements. These statements are constructed using words and these words are constructed using characters from C character set. C language character set contains the following set of characters...

1. Alphabets
2. Digits
3. Special Symbols

Alphabets

C language supports all the alphabets from the English language. Lower and upper case letters together support 52 alphabets.

lower case letters - **a to z**

UPPER CASE LETTERS - **A to Z**

Digits

C language supports 10 digits which are used to construct numerical values in C language.

Digits - **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**

Special Symbols

C language supports a rich set of special symbols that include symbols to perform mathematical operations, to check conditions, white spaces, backspaces, and other special symbols.

Special Symbols - **~ @ # $ % ^ & * ( ) _ - + = { } [ ] ; : ' " / ? . > , < \ | tab newline space NULL bell backspace verticaltab etc.,**

Whitespace Characters

These are employed at the time of execution of the program. Execution characters set are always represented by a backslash (\) followed by a character. Note that each one of character constants represents one character, although they consist of two characters. These characters combinations are called as **escape sequence**.

\b	blank space	\f	form feed	\"	Double quote
\t	horizontal tab	\n	new line	\?	Question mark
\v	vertical tab	\\	Back slash	\0	Null
\r	carriage return	\’	Single quote	\a	Alarm (bell)

C Keywords

Keywords are predefined, reserved words used in programming that have special meanings to the compiler. Keywords are part of the syntax and they cannot be used as an identifier. For example:

Int money;

Here, int is a keyword that indicates money is a variable of type int (integer).
As C is a case sensitive language, all keywords must be written in lowercase. Here is a list of all keywords allowed in ANSI C.

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

Variable

What is a Variable?

A variable is an identifier which is used to store some value. Constants can never change at the time of execution. Variables can change during the execution of a program and update the value stored inside it.

A single variable can be used at multiple locations in a program. A variable name must be meaningful. It should represent the purpose of the variable.

In programming, a variable is a container (storage area) to hold data.

To indicate the storage area, each variable should be given a unique name (identifier). Variable names are just the symbolic representation of a memory location. For example:

```
int playerScore = 95;
```

Here, *playerScore* is a variable of int type. Here, the variable is assigned an integer value 95.

Rules for naming a variable

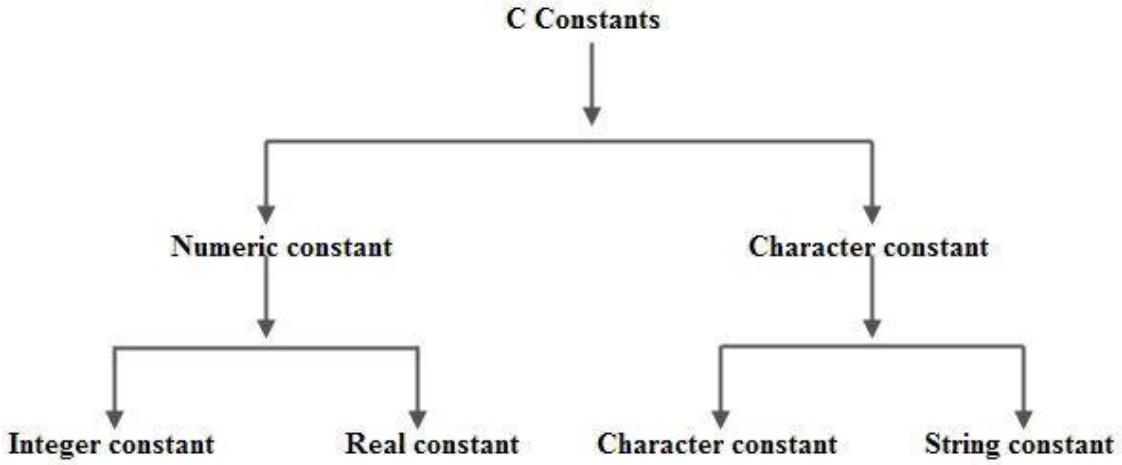
1. A variable name can only have letters (both uppercase and lowercase letters), digits and underscore.
2. The first letter of a variable should be either a letter or an underscore.
3. There is no rule on how long a variable name (identifier) can be. However, you may run into problems in some compilers if the variable name is longer than 31 characters.
4. A variable name should not begin with a number.
5. A variable name should not consist of whitespace.
6. A variable name should not consist of a keyword.
7. 'C' is a case sensitive language that means a variable named 'age' and 'AGE' are different.

Constant

What is Constants? Type of constant

There are many different types of data values that are implicitly declared as constants in C. The value of a constant cannot be changed during execution of the program, neither by the programmer nor by the computer. The character 'A' is a constant having numerical value equal to 65 in decimal number system.
Similarly 'B', 'C', etc., are other constant values, for instance, 'B'= 66, 'C' = 67, etc. In any C program, the value of character 'A' cannot be changed. Similarly, a sequence of characters enclosed between double quotes such as "Morning" is a string constant. Also, the characters "\n" and "\t' are constants, which have special meaning for compiler, and are known as escape sequences.

C supports several **types of**



Integer Constants

An integer constant is a sequence of digits from 0 to 9 without decimal points or fractional part or any other symbols. There are 3 types of integers namely decimal integer, octal integers and hexadecimal integer.

Decimal Integers consists of a set of digits 0 to 9 preceded by an optional + or – sign. Spaces, commas and non digit characters are not permitted between digits. Example for valid decimal integer constants are

int y=123; //here 123 is a decimal integer constant

Real Constants

Real Constants consists of a fractional part in their representation. Integer constants are inadequate to represent quantities that vary continuously. These quantities are represented by numbers containing fractional parts like 26.082. Example of real constants are

float x = 6.3; //here 6.3 is a double constant.
float y = 6.3f; //here 6.3f is a float constant.
float z = 6.3 e + 2; //here 6.3 e + 2 is a exponential constant.
float s = 6.3L ; //here 6.3L is a long double constant

Real Numbers can also be represented by exponential notation. The general form for exponential notation is mantissa exponent. The mantissa is either a real number expressed in decimal notation or an integer. The exponent is an integer number with an optional plus or minus sign.

Single Character Constants

A Single Character constant represent a single character which is enclosed in a pair of quotation symbols.

Example for character constants are

char p =’ok’ ; // p will hold the value ‘O’ and k will be omitted
char y =’u’; // y will hold the value ‘u’
char k =’34’ ; // k will hold the value ‘3, and ‘4’ will be omitted
char e =’ ‘; // e will hold the value ‘ ‘ , a blank space
chars =’\45’; // swill hold the value ‘ ‘ , a blank space

All character constants have an equivalent integer value which are called ASCII Values.

String Constants

A string constant is a set of characters enclosed in double quotation marks. The characters in a string constant sequence may be a alphabet, number, special character and blank space. Example of string constants are

“VISHAL” “1234” “God Bless” “!…..?”

Backslash Character Constants [Escape Sequences]

Backslash character constants are special characters used in output functions. Although they contain two characters they represent only one character. Given below is the table of escape sequence and their meanings.

Constant	Meaning
‘\a’	.Audible Alert (Bell)
‘\b’	.Backspace
‘\f’	.Formfeed

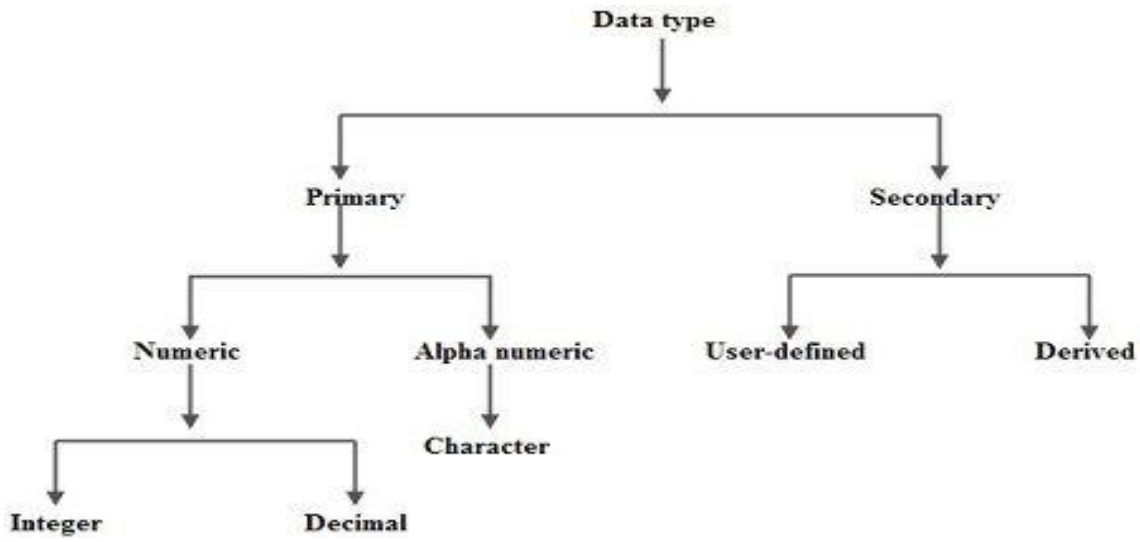
'\n'	.New Line
'\r'	.Carriage Return
'\t'	.Horizontal tab
'\v'	.Vertical Tab
'\''	.Single Quote
'\"'	.Double Quote
'\?'	.Question Mark
'\\'	.Back Slash
'\0'	.Null

Data Types in c

C language supports wide varieties of data types to accommodate any types of data manipulation the variety of data types available allow the programmer to select the type appropriate to the – needs for the program as well as the machine. Mainly the C language supports two types of data. type such as:

1. Primary data types (or the fundamental data types).
2. Secondary data types.

Again the secondary data type is divided into two types one is Derived data type and another one is User-defined data type. The total graphical representation of data types is given in the Fig.



In C programming, data types are declarations for variables. This determines the type and size of data associated with variables. For example,

```
int myvar;
```

Here, *myVar* is a variable of int (integer) type. The size of int is 4 bytes.

Basic types

Here's a table containing commonly used types in C programming for quick access.

Type	Size (bytes)	Format Specifier
int	at least 2, usually 4	%d, %i
char	1	%c
float	4	%f
double	8	%lf
short int	2 usually	%hd
unsigned int	at least 2, usually 4	%u
long int	at least 4, usually 8	%ld, %li
long long int	at least 8	%lld, %lli
unsigned long int	at least 4	%lu
unsigned long long int	at least 8	%llu
signed char	1	%c
unsigned char	1	%c
long double	at least 10, usually 12 or 16	%Lf

int

Integers are whole numbers that can have both zero, positive and negative values but no decimal values. For example, 0, -5, 10
We can use int for declaring an integer variable.

```
Int id;
```

Here, *id* is a variable of type integer.
You can declare multiple variables at once in C programming. For example,

The size of int is usually 4 bytes (32 bits). And, it can take 2^{32} distinct states from -2147483648 to 2147483647.

float and double

float and double are used to hold real numbers.

```
Float salary;
Double price;
```

In C, floating-point numbers can also be represented in exponential. For example,

```
float normalizationFactor = 22.442e2;
```

The size of `float` (single precision float data type) is 4 bytes.
And the size of `double` (double precision float data type) is 8 bytes.

char
Keyword `char` is used for declaring character type variables. For example,

```
char test = 'h';
```

The size of the character variable is 1 byte.

void

`void` is an incomplete type. It means "nothing" or "no type". You can think of void as **absent**.

For example, if a function is not returning anything, its return type should be `void`.

Note that, you cannot create variables of `void` type.

https://ecomputernotes.com/what-is-c/types-and-variables/data-types-in-c

Declaration of Variables

Every variable used in the program should be declared to the compiler. The declaration does two things.

• Tells the compiler the variables name.

• Specifies what type of data the variable will hold.

The general format of any declaration

datatype v1, v2, v3,.... vn;
Where v1, v2, v3 are variable names. Variables are separated by commas. A declaration statement must end with a semicolon.

Structure of C programm

There are six main sections to a basic c program.

The six sections are,

- Documentation
- Link
- Definition
- Global Declarations
- Main functions
- Subprograms

So now that the introduction is out of the way, let us jump to the main discussion. The whole code follows this outline. Each code has a similar outline. Now let us learn about each of this layer in detail.

Figure: Basic Structure Of C Program

Documentation Section
The Documentation Section consists of a set of comment lines giving the name of the Programmer, date and other details about the program. Documentation section helps anyone to get an overview of the program. Comments may appear anywhere within a program. Text between **/*** And ***/** appears as a comment in C.
for Example: **/* This is a Comment */**

Link Section
This part of the code is used to declare all the header files that will be used in the program. This leads to the compiler being told to link the header files to the system libraries.
Example
1#include<stdio.h>

Definition Section
In this section, we define different constants. The keyword define is used in this part.
1#define PI=3.14

Global Declaration Section
This part of the code is the part where the global variables are declared. All the global variable used are declared in this part. The user-defined functions are also declared in this part of the code.
Int x;

Main Function Section
Every C-programs needs to have the main function. Each main function contains 2 parts. A declaration part and an Execution part. The declaration part is the part where all the variables are declared. The execution part begins with the curly brackets and ends with the curly close bracket. Both the declaration and execution part are inside the curly braces.

```
Main()
{
Int a=10;
Printf(“”%d”,a);
}
```

Sub Program Section
All the user-defined functions are defined in this section of the program.