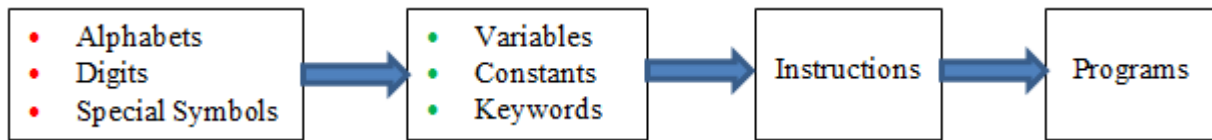


STEPS IN LEARNING C



One must understand character set used while writing C programs. Characters in C are grouped into the following categories:

1. **Letters or Alphabets**
A, B, C,Z
a, b, c,z

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

3. **Special symbols or characters**
+ - * # \$ @ ? . , { ([]) } etc

4. **White spaces:** also known as **blanks** or **blank spaces**



Keyboard

Note: A character set denotes any English letter (A, B, C,Z or a,b,c.....z), digits (0,1,2.....9) or special symbols (+, -, *, /, #, \$ etc) used to represent information.

VARIABLES

Variables are used to store values. We can also define variables as identifiers i.e., names used to hold values.

RULES FOR NAMING VARIABLES

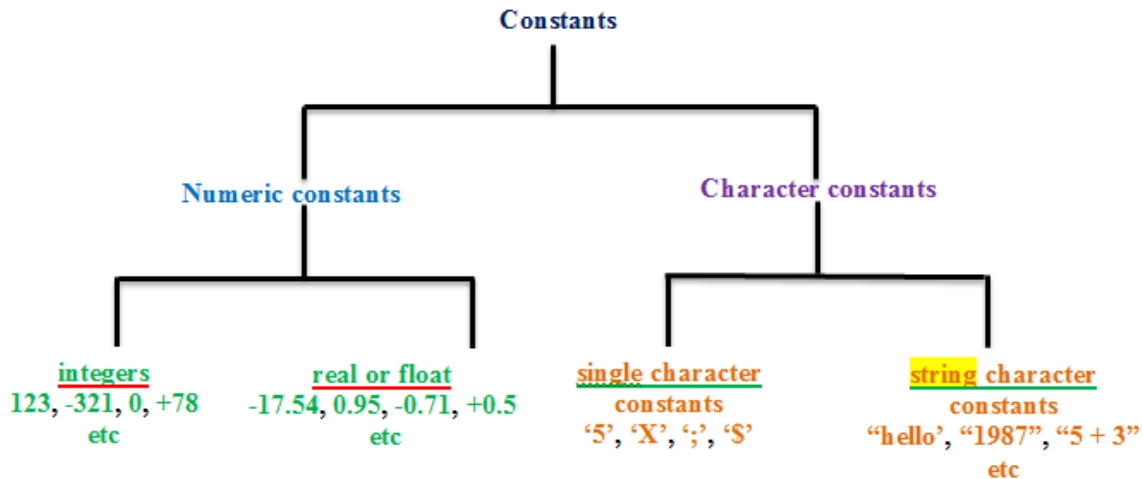
1. First character must be an alphabet [English Letter] or **underscore symbol** (_).
2. Variable names must not contain white spaces [**BLANKS**]
3. First character may be followed by a sequence of letters and digits
4. **Keywords** (also known as **reserved words**) must not be used as variable names. There are **32** keywords in C language – **int, float, char, for, while, do const, struct, union etc.**
5. No special characters except underscore is used in variable names.
6. In C, uppercase and lowercase letters are different. Therefore the variable name **SUM, Sum, sum** are different variables. **Good practice is to represent variable names in LOWER-CASE.**

Note: It is useful to give meaningful names to the variables.

Valid names	Invalid names
Average	5bc
height	int
total	float
net_pay	gopal\$
_gopal	gopal 123
John	John
Delhi	Delhi#
a	
b	

CONSTANTS

Constants in **C** refer to fixed values. They do not change during execution of a program.



Characters must be enclosed in single quotes



Strings must be enclosed in double quotes

- ⊗ Single character constants are enclosed in single quotes.
- ⊗ String character constants are enclosed in double quotes.
- ⊗ String = collection of characters
- ⊗ Integers do not contain fractions.
- ⊗ Real constants may contain fraction values.

KEYWORDS

There are 32 keywords available in C.

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

N.B. keywords have special meaning and purpose

scanf AND printf FUNCTIONS

scanf()



scanf function is used to supply input to the program from standard input device KEYBOARD. scanf() receives values from keyboard. Format of scanf function is as follows:

```
scanf("control string", variable1, variable2, ....);
```

Note that ampersand (&) symbol must be used before the variable names in scanf(). Ampersand is known as "address of operator".

```
scanf("%d %d %d", &a, &b, &c);
```

here a, b, c are variable names of type int.

Note: scanf() function is counterpart of printf function.

printf()



printf function is used to print a message or value on the screen. SCREEN is a standard output device. printf() sends values to screen. General form of printf() function is as follows:

```
printf("control string", variable1, variable2, ....);
```

printf() has two forms:

1. Simply sends text in double quotes to screen

```
printf("GOPALAAAAAAAA");
```

```
printf("control string")
```

format string

2. Sends values in the program to screen

```
printf("%d %f %d", a, b, c);
```

here values of variables a,b,c are printed on screen.

CONVERSION SPECIFICATIONS

Conversion specifiers tells the scanf() and printf() functions about the type of data (values) to be handled.

Conversion specifier	Meaning
%d	integer type value
%f	real or float value
%c	character value
%s	string value
%u	unsigned value

unsigned means only positive value

FORMAT OF SIMPLE C PROGRAM

```
#include<stdio.h> ← #include<filename>
void main() ← function name
{ ← start of program
    .....
    ..... ← program statements
    .....
    .....
} ← end of program
```

Note

#include is a pre-processor directive and is written as #include<filename>. Filename is the name of the library file that contains the required function definition.

SIMPLE PROGRAMS

Program 1: Hello World

```
#include<stdio.h>

void main()
{
printf("Hello World");
}
```

Program 1: Sum of given two numbers (inputs are directly coded in the program)

```
#include<stdio.h>

void main()
{
    int a, b, sum; /*variable declaration*/
    a=10;
    b=20;
    sum = a+b;

    printf("%d", sum);
}
```

Sum of given two numbers (using scanf)

```
#include<stdio.h>

void main()
{
    int a, b, c, sum; /*variable declaration*/
    printf("Enter value of a: ");
    scanf("%d", &a);

    printf("Enter value of b: ");
    scanf("%d", &b);
    sum = a+b;

    printf("%d", sum);
}
```

- Count the number of variables required in the program (in this program 3 variables required)
- Above program requires 2 input variables (a, b) and 1 output variable sum
- Declare type of variables. Here a, b and sum are declared as int type
- Decide how to give input to the program – either direct initialization or using scanf

Program 2: Calculation of simple interest (inputs are directly coded in the program)

```
#include<stdio.h>

void main()
{
    int p, n; /*variable declaration */
    float r, si; /*variable declaration */
    p = 31000;
    n = 3;
    r = 8.5;
    si = (p*n*r)/100;

    printf("simple interest is %f", si);
}
```

Calculation of simple interest (using scanf)

```
#include<stdio.h>

void main()
{
    int p, n; /*variable declaration */
    float r, si; /*variable declaration */

    printf("Enter values of p, n, r: ");
    scanf("%d %d %f", &p, &n, &r);

    si = (p*n*r)/100;
    printf("simple interest is %f", si);
}
```

Program 3: calculation of average of 3 numbers
(inputs are directly coded in the program)

```
#include<stdio.h>

void main()
{
    int a, b, c; /*variable declaration*/
    float average; /*variable declaration*/
    a = 10;
    b = 20;
    c = 30;
    average = (a+b+c)/3;

    printf("%f", average);
}
```

calculation of average of 3 numbers
(using scanf)

```
#include<stdio.h>

void main()
{
    int a, b, c; //variable declaration
    float average;

    printf("Enter inputs: ");
    scanf("%d %d %d", &a, &b, &c);

    average = (a+b+c)/3;

    printf("%f", average);
}
```

Program 4

Suresh basic salary is input through the keyboard. His DA is 40% of basic salary and HRA is 20% of basic salary. Write a C program to calculate his gross salary.

```
#include<stdio.h>

void main()
{
    int basic;
    float DA;
    float HRA;
    float gross;
    printf("Enter basic salary: ");
    scanf("%d", &basic);
    DA = (basic * 40)/100;
    HRA = (basic * 20)/100;

    gross = basic + DA + HRA;

    printf("Gross = %f", gross);
}
```

```
#include<stdio.h>

void main()
{
    int basic;
    float DA;
    float HRA;
    float gross;

    basic = 45000;

    DA = (basic * 40)/100;
    HRA = (basic * 20)/100;
    gross = basic + DA + HRA;

    printf("Gross = %f", gross);
}
```

Program 5

If the marks obtained by a student in 5 different subjects are input through the keyboard, find the aggregate marks and % of marks obtained by the student. Assume that the maximum marks that can be obtained by a student in each subject is 100.

Program 5: Calculation of percentage (inputs are directly coded in the program)

```
#include <stdio.h>

void main()
{
    int m1, m2, m3, m4, m5;
    int total;
    float percent;

    m1 = 50;
    m2 = 60;
    m3 = 70;
    m4 = 80;
    m5 = 90;

    total = m1 + m2 + m3 + m4 + m5;
    percent = (total * 100)/500;

    printf("Total marks = %d\n", total);
    printf("percentage obtained = %f", percent);
}
```

Calculation of percentage (using scanf)

```
#include<stdio.h>

id main()

    int m1, m2, m3, m4, m5;
    int total;
    float percent;
    printf("Enter marks in 5 subjects: ");
    scanf("%d %d %d %d %d", &m1, &m2, &m3, &m4, &m5);

    total = m1 + m2 + m3 + m4 + m5;
    percent = (total * 100)/500;

    printf("Total marks = %d\n", total);
    printf("percentage obtained = %f", percent);
```

EXERCISES

- Using arithmetic operators in C, write a program that declares and assigns values to the variables **a**, **b**, **c** and then does the following:
 - Halves the value of **a**
 - Doubles **b**
 - Multiplies **c** by itself
 - Prints out the results of above operations
- Enter the temperature in Celsius and convert that into Fahrenheit. Formula: $F = \frac{9}{5} * C + 32$;
- Two variables **a** and **b** contain values of 10 and 20. Write a program to interchange the contents of **a** and **b**.
- Write a program to find area of a triangle. Input the length and breadth of a rectangle through the keyboard. Formula: $Area = length * breadth$
- Write a program to find area of a circle of any radius (r). Formula: $Area = \pi r^2$

C/C++ compilers

- ☛ Bloodshed Dev C++ compiler
- ☛ Turbo C++
- ☛ Borland C/C++
- ☛ Quincy
- ☛ Code Blocks
- ☛ Digital Mars C/C++