Department of Computer Science and Engineering,
PES University, Bangalore, India

Lecture Notes
Problem Solving With C
UE24CS151B

Lecture #6
Simple Input function in C

By,
Prof. Sindhu R Pai,
Theory Anchor, Feb-May, 2025
Assistant Professor
Dept. of CSE, PESU

Many Thanks to
Dr. Shylaja S S (Director, CCBD and CDSAML Research Center, PES University)
Prof. Nitin V Poojari (Dean, Internal Quality Assurance Cell, PES University)

Unit #: 1

Unit Name: Problem Solving Fundamentals

Topic: Simple Input functions in C

Course objectives: The objective(s) of this course is to make students

- Acquire knowledge on how to solve relevant and logical problems using computing Machine.
- Map algorithmic solutions to relevant features of C programming language constructs.
- Gain knowledge about C constructs and its associated ecosystem.
- Appreciate and gain knowledge about the issues with C Standards and it's respective behaviours.

Course outcomes: At the end of the course, the student will be able to:

- Understand and Apply algorithmic solutions to counting problems using appropriate C Constructs.
- Understand, Analyze and Apply sorting and Searching techniques.
- Understand, Analyze and Apply text processing and string manipulation methods using Arrays, Pointers and functions.
- Understand user defined type creation and implement the same using C structures, unions and other ways by reading and storing the data in secondary systems which are portable.

Sindhu R Pai

Theory Anchor, Feb - May, 2025

Dept. of CSE,

PES University

Input and Output functions in C

Input and output functions are available in the c language to perform the most common tasks.

In every c program, three **basic functions take place namely accepting of data as input, the processing of data, and the generation of output**.

When a programmer says input, it would mean that they are feeding some data in the program. Programmer can give this input from the command line or in the form of any file. The c programming language comes with a set of various built-in functions for reading the input and then feeding it to the available program as per our requirements.

When a programmer says output, they mean displaying some data and information on the printer, the screen, or any other file. The c programming language comes with various built-in functions for generating the output of the data on any screen or printer, and also redirecting the output in the form of binary files or text file.

Simple Input functions in C

Two Types: Formatted Input function and Unformatted Input function

Formatted input function – scanf

scanf takes a format string as the first argument. The input should match this string**.**

int scanf(const char *format, ...);

where

int (integer) is the return type

format is a stringthatcontains the type specifier(s)

"..." (ellipsis) indicates that thefunction accepts a variable number of arguments; each argument must be amemory address where the converted result is written to. On success, the function writesthe result into the arguments passed.

- scanf("%d%d", &var1,&var2);// & -address operator is compulsory in scanf for all primary types

  When the user types, 23 11, 23 is written to var1 and 11 is to var2.

- scanf("%d,%d", &a, &b);

  scanf has a comma between two format specifiers, the input should have a commabetween a pair of integers.

- scanf("%d%d\n", &a, &b);

  It is not a good practice to have any kind of escape sequence in scanf. In the above code, it expects two integers. Ex: 20 30. Then if you press enter key, scanf does not terminate. You need to give some character other than the white space.

The function returns the following value:

>0 - The number of items converted and assigned successfully.

0 - No item was assigned.

<0 - Read error encountered or end-of-file (EOF) reached before any assignment was made.

- n = scanf("%d",&a);// If user enters 20, a becomes 20 and 1 is returned by the function.
- n = scanf("%d,%d",&a,&b); //If user enters 20 30, a becomes 20, value of b is undefinedand 1 is returned by the function

Unformatted input functions

A character in programming refers to a single symbol. A character in 'C' is like a very small integer having one of the 256 possible values. It occupies a single byte. We code English alphabets A as 65, B as 66, Z as 90, a as 97 and so on in a coding scheme called ASCII. To read a character from the keyboard, we could use

scanf("%c", x);We could also use **x = getchar();**

We prefer the second as it is developed only for handling a single char and therefore more efficient even though it is not generic. The unformatted input functions further have two categories: **Character functions and string functions**

Character functions: getchar(), getche(), getch()

String functions: gets() – This will be discussed in Unit - 2

Coding Example_1: To read two characters and display them.

```
int main()
{
        char ch = 'p'; // ch is a variable of type char and it can store only one character at a
        time.
        //Value for a character variable must be within single
        quote. // printf("Ch is %c\n",ch);// p
        //char ch1 = 'pqrs';// TERRIBLE
        CODEprintf("Ch is %c\n",ch);// s
        /*
        char x; char y;
        scanf("%c", &x); scanf("%c", &y);
        printf("x:%c y:%c\n", x, y);
        */
        x = getchar(); y = getchar(); putchar(x);
        putchar(y); printf("%d", y);
        return 0;
}
```

// If I enter P<newline>, x becomes P and y becomes newline

// scanf and printf : generic; not simple

// getchar and putchar : not generic; read /write a char; simple

// If I enter p<space>q, q will not be stored in y. Only space is stored. This can be avoided using fflush(stdin) function between two getchar function calls. This function clears the key board buffer.

// fflush(stdin) –windows

//_fpurge(stdin) – Linux based. Include <stdio_ext.h> in Linux based

Note: While using scanf for reading character input, care should be taken to handle White Spaces and Special Characters handling buffering problem.

Happy Coding using Input functions in C!!