

PF LAB # 08

Strings and Structures



Romasha Khurshid

Nida Munawwar

# Learning Objectives

This lab will cover the following topics:

* Strings
* Structures

**C Strings:**

* The string can be defined as the one-dimensional array of characters terminated by a null ('\0').
* The character array or the string is used to manipulate text such as word or sentences.
* Each character in the array occupies one byte of memory, and the last character must always be 0.
* The termination character ('\0') is important in a string since it is the only way to identify where the string ends.
* When we define a string as char s[10], the character s[10] is implicitly initialized with the null in the memory.

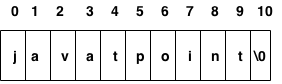
There are two ways to declare a string in c language.

1. **By char array**
2. **By string literal**

Let's see the example of declaring **string by char array** in C language

**char** ch [10] = {'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't', '\0'};

As we know, array index starts from 0, so it will be represented as in the figure given below.



While declaring string, size is not mandatory. So we can write the above code as given below:

**char** ch[]={'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't', '\0'};

We can also define the **string by the string literal** in C language. For example:

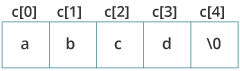
**char** ch[]="javatpoint";

In such case, '\0' will be appended at the end of the string by the compiler.

## How to initialize strings?

You can initialize strings in a number of ways.

1. char c[] = "abcd";
2. char c[50] = "abcd";
3. char c[] = {'a', 'b', 'c', 'd', '\0'};
4. char c[5] = {'a', 'b', 'c', 'd', '\0'};



Let's take another example:

1. char c[5] = "abcde";

Here, we are trying to assign 6 characters (the last character is '\0') to a char array having 5 characters. This is bad and you should never do this.

## Read String from the user

You can use the scanf() function to read a string.

The scanf() function reads the sequence of characters until it encounters [whitespace](https://stackoverflow.com/questions/30033582/what-is-the-symbol-for-whitespace-in-c) (space, newline, tab etc.).

### **Example 1: scanf() to read a string**

1. #include <stdio.h>
2. int main()
3. {
4. char name[20];
5. printf("Enter name: ");
6. scanf("%s", name);
7. printf("Your name is %s.", name);
8. return 0;
9. }

**Output**

Enter name: Dennis Ritchie

Your name is Dennis.

Even though Dennis Ritchie was entered in the above program, only "Ritchie" was stored in the name string. It's because there was a space after Dennis.

### scanf( ) is not capable of receiving multi-word strings. Therefore names such as ‘Dennis Ritchie would be unacceptable.

### The way to get around this limitation is by using the function gets( ).

### The usage of functions gets( ) and its counterpart puts( ) is shown below.

**Code:**

#include <String.h>

### main( )

### {

### char name[25] ;

### printf ( "Enter your full name " ) ;

### gets ( name ) ;

### puts ( "Hello!" ) ;

### puts ( name ) ;

### }

### **And here is the output...**

### Enter your name Dennis Ritchie

### Hello!

### Dennis Ritchie

**Standard Library String Functions:**

With every C compiler a large set of useful string handling library functions are provided. Given below are more commonly used functions along with their purpose.



## 

## 

## 

## 

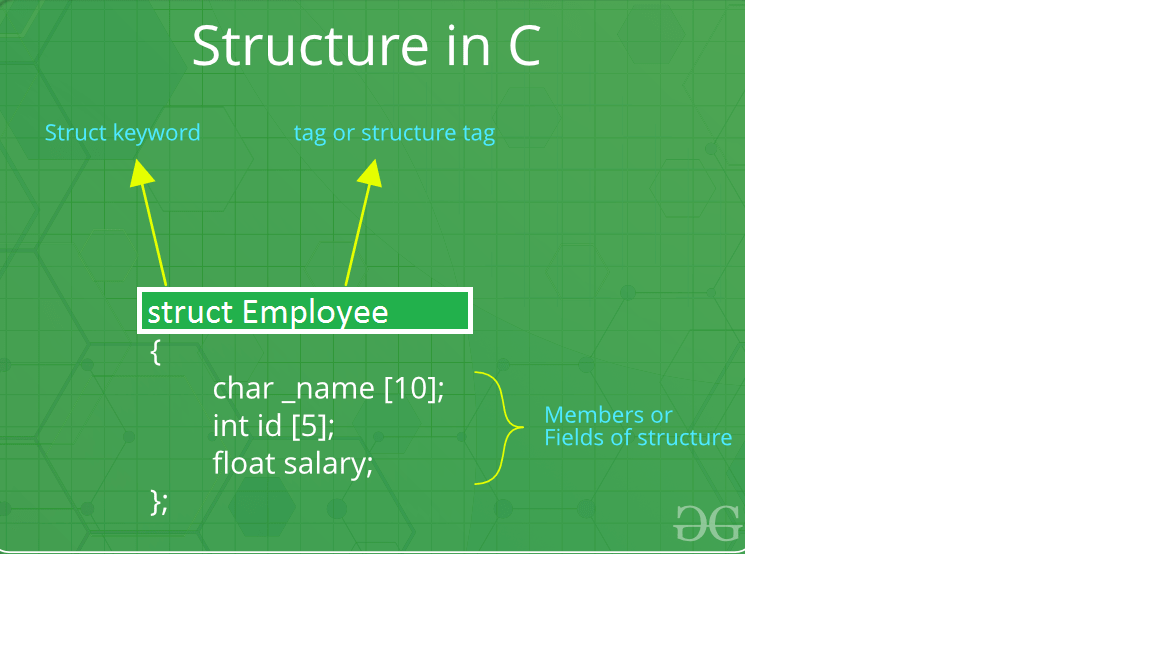
## 

## 

## 

## 

**Structures in C:**

****

|  |
| --- |
| **What is a structure?** A structure is a user defined data type in C/C++. A structure creates a data type that can be used to group items of possibly different types into a single type.  **How to create a structure?** ‘struct’ keyword is used to create a structure. Following is an example.  struct address  {     char name[50];     char street[100];     char city[50];     char state[20];     int pin;  }; |

**How to declare structure variables?**

A structure variable can either be declared with structure declaration or as a separate declaration like basic types.

|  |
| --- |
| // A variable declaration with structure declaration.  struct Point  {     int x, y;  } p1; // The variable p1 is declared with 'Point'      // A variable declaration like basic data types  struct Point  {     int x, y;  };    int main()  {     struct Point p1;  // The variable p1 is declared like a normal variable  } |

Note: In C++, the struct keyword is optional before in declaration of a variable. In C, it is mandatory.

**How to initialize structure members?**

Structure members **cannot be** initialized with declaration. For example the following C program fails in compilation.

|  |
| --- |
| struct Point  {     int x = 0;  // COMPILER ERROR:  cannot initialize members here     int y = 0;  // COMPILER ERROR:  cannot initialize members here  }; |

The reason for above error is simple, when a datatype is declared, no memory is allocated for it. Memory is allocated only when variables are created.

Structure members **can be** initialized using curly braces ‘{}’. For example, following is a valid initialization.

|  |
| --- |
| struct Point  {     int x, y;  };    int main()  {     // A valid initialization. member x gets value 0 and y     // gets value 1.  The order of declaration is followed.     struct Point p1 = {0, 1};  } |

**How to access structure elements?**

Structure members are accessed using dot (.) operator.

|  |
| --- |
| #include<stdio.h>    struct Point  {     int x, y;  };    int main()  {     struct Point p1 = {0, 1};       // Accesing members of point p1     p1.x = 20;     printf ("x = %d, y = %d", p1.x, p1.y);       return 0;  } |

**Output:**

x = 20, y = 1

**What is designated Initialization?**

Designated Initialization allows structure members to be initialized in any order. This feature has been added in [C99 standard](https://www.geeksforgeeks.org/c-programming-language-standard/).

|  |
| --- |
| #include<stdio.h>    struct Point  {     int x, y, z;  };    int main()  {     // Examples of initializtion using designated initialization     struct Point p1 = {.y = 0, .z = 1, .x = 2};     struct Point p2 = {.x = 20};       printf ("x = %d, y = %d, z = %d\n", p1.x, p1.y, p1.z);     printf ("x = %d", p2.x);     return 0;  } |

**Output:**

x = 2, y = 0, z = 1

x = 20

This feature is not available in C++ and works only in C.

# Structures as Function Arguments

You can pass a structure as a function argument in the same way as you pass any other variable or pointer.

#include <stdio.h>

#include <string.h>

struct Books {

char title[50];

char author[50];

char subject[100];

int book\_id;

};

/\* function declaration \*/

void printBook( struct Books book );

int main( ) {

struct Books Book1; /\* Declare Book1 of type Book \*/

struct Books Book2; /\* Declare Book2 of type Book \*/

/\* book 1 specification \*/

strcpy( Book1.title, "C Programming");

strcpy( Book1.author, "Nuha Ali");

strcpy( Book1.subject, "C Programming Tutorial");

Book1.book\_id = 6495407;

/\* print Book1 info \*/

printBook( Book1 );

return 0;

}

void printBook( struct Books book ) {

printf( "Book title : %s\n", book.title);

printf( "Book author : %s\n", book.author);

printf( "Book subject : %s\n", book.subject);

printf( "Book book\_id : %d\n", book.book\_id);

}

**Lab Activities**

1. Write a program to reverse a string using “strrev(string)” function in C.
2. Write a program to reverse a string without using “strrev” function in C.
3. Write a program to delete all vowels from a sentence. Assume that the sentence is not more than 60 characters long.
4. Write a program to count the length of string without built in function.
5. Write a program that extracts part of the given string from the specified position. For example if the string is “Working with string is fun”, then it from position 4, 4 character are to be extracted then program should return string as a “ing…………….
6. Write a C program to print the record of 3 students using structure.
7. Define a struct type with the name Length that represents a length in yards, feet, and inches. Define an add() function that will add two Length arguments and return the sum as type Length. Define a second function, show(), that will display the value of its Length argument. Write a program that will use the Length type and the add() and show() functions to sum an arbitrary number of lengths in yards, feet.
8. Write a program that generates and prints the fibbonacci words of order 0 through 5. If f(0)=’a’, f(1)=’b’, f(2)=”ba”, f(3)=”bab”, f(4)=”babba” etc.
9. Write a program that takes a set of names of individuals and abbreviates the first, middle and other names except the last name by their first letter.
10. Write a program to count the number of occurrences of any two vowels in succession in a line of text. For example, in the sentence “Pleases read this application and give me gratuity” such occurrences are ea, ea, ui.