Programming Fundamentals (CT-175)
Lab 09

User Defined Functions – Declaration, definition, and function call

Objectives

The objective of this lab is to enable students create user defined functions by passing and returning different types of arguments. By the end of this lab students will be able to write user defined functions in C.

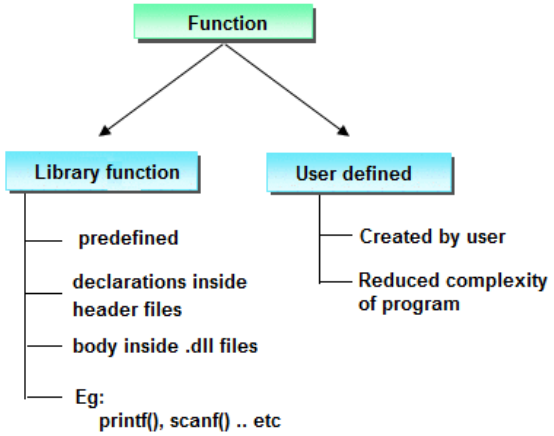
Tools Required

DevC++ IDE

Course Coordinator –
Course Instructor –
Lab Instructor –
Prepared By Department of Computer Science and Information Technology
NED University of Engineering and Technology

Functions in C

A large program can be divided into basic building blocks, for performing specific tasks, known as functions. A function contains the set of programming statements enclosed by curly braces { }. The function is also known as procedure or subroutine in other programming languages. Functions are mainly classified into two categories, that is, library and user defined functions.



Following are a few benefits of dividing a large program in to small units called functions.

- ✓ It provides modularity to your program's structure.
- ✓ It makes your code reusable. You just have to call the function by its name to use it, wherever and whenever required.
- ✓ In case of large programs with thousands of code lines, debugging and editing becomes easier if you use functions.
- ✓ It makes the program more readable and easy to understand.

User Defined Functions

There are three steps for defining user defined functions in C, that is, function declaration, definition, and call.

Function Declaration or Function Prototype

A function prototype is simply the declaration of a function that specifies function's name, parameters and return type. It doesn't contain function body. A function prototype gives information to the compiler that the function may later be used in the program. The function prototype is not needed if the user-defined function is defined before the main( ) function. Syntax for function declaration is as follows.

```
return_data_type functionName(data_type1 argument1, data_type2 argument2,...);
int add(int a, int b);
```

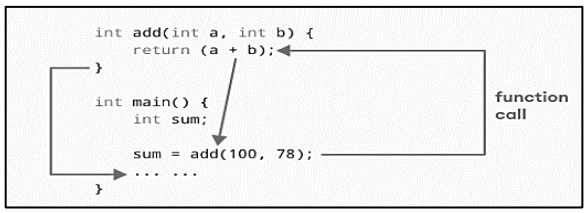
Function Definition

Function definition contains the block of code to perform a specific task and returning the result. Syntax for function definition is as follows.

```
return data_type functionName(data_type1 argument1, data_type2 argument2,{
//body of the function
}
```

Function Call

When a function is called, the control of the program is transferred to the function definition. And, the compiler starts executing the codes inside the body of a function. In programming argument refers to the variable passed to the function.



In the above example, we have function definition and functions call which provides following information to the compiler:

- ✓ name of the function is add()
- ✓ return type of the function is int
- ✓ two arguments (100, 78) of type int are passed to the function. The parameters a and b accepts the arguments passed in the function definition. These arguments are called formal parameters of the function and store the copy of actual parameters.
- ✓ The return statement terminates the execution of a function and returns a value to the calling function. The program control is transferred to the calling function after return statement. In the above example, we have the data type int instead of void. This means that the function returns an int value.
- ✓ In the above example, function call is made using add(100,78); statement inside the main().

User Defined Head Files

The purpose to understand and learn header file is that, it also contains specific function defines in it. You are required to call its header and can use its defined function in your program. Header file serves two purposes.

- ✓ You include them in your program to supply the definitions and declarations you need to invoke system calls and libraries.
- ✓ Your own header files contain declarations for interfaces between the source files ofyour program. Each time you have a group of related declarations and macro definitions all or most of which are needed in several different source files.

Follow the step to create your header file.

- ✓ Make a header file with .h extension and give it unique name e.g sumfile-> sumfile.h
- ✓ Define your program in header extension file.
  - int add(int a,int b){
    return(a+b);
    }
- ✓ Make source file of c, where your main program is set.

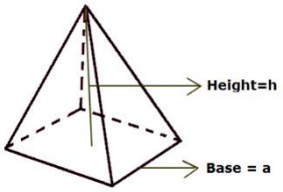
```
#include<stdio.h>
#include "sumfile.h" // don't use '<>', instead of it use"".
void main(){ int num1 = 10, num2 = 10, num3;
num3 = add(num1, num2);
printf("Addition of Two numbers : %d", num3);
}
```

- ✓ Keep .h file path directory same as source file.
- ✓ In the above program the 'add' function is basically called from the heder file of sumfile.h which we have explicitly defined.

Exercise

1. Write a function that prints all the unique values from an array and the number of times each value occurred. The main function takes a size of array as input and generates a random integer array with name **"array1"**. Random number limit must be between 0 and 10. The 'main' function calls a function with the name as "CountFrequency( )" that will find the occurrence of each value in array.
2. Salesflow is one of leading software house they are starting their recruitment process for three different following positions: Associate Developer, Assistant Developer, Trainee Engineer. There is a defined criterion for recruitment process: if candidate clears the test with 50 marks, he will be selected for the post of trainee engineer; experience is not the required for this post. If candidate secures 60 marks with at least one year of experience and 70 marks with at least 2 years of experience, then he/she will be selected as an assistant and associate developer, respectively. Write a function that takes the test marks from user and ask for experience (if the entered marks are x >=60). After that, function shows the assigned position.
3. Write the program that calculate the volume by using following formula

$$V = a\ 2 * 1\ /3\ h,$$



by creating two separate functions. One of the functions with prototype "getData(int h, int a)", takes two inputs from user. The other function with prototype "volumeCal( )" calculates the volume, and this function must be called from the first function "getData ( )". The first function must be called from the main function.

4. Write a program that takes a positive number with a fractional part and rounds it to two decimal places. For example, 32.4851 would round to 32.49, and 32.4431 would round to 32.44.
5. In shopping for a new house, you must consider several factors. In this problem the initial cost of the house, the estimated annual fuel costs, and the annual tax rate are available. Write a program that will determine the total cost of a house after a five-year period and run the program for each of the following sets of data.

Initial House Cost	Annual Fuel Cost	Tax Rate
67,000	2,300	0.025
62,000	2,500	0.025
75,000	1,850	0.020

To calculate the house cost, add the initial cost to the fuel cost for five years, then add the taxes for five years. Taxes for one year are computed by multiplying the tax rate by the initial cost. Write and call a function that displays instructions to the program user.

Lab 09 Evaluation		
Student Name:	**Student ID:**	**Date:**
Task No.	**Marks**	**Remarks by teacher in accordance with the rubrics**
1		
2		
3		
4		
5		