Introduction

In this lab session, you'll learn about how to write a function (or a method) and how to invoke a function properly in C.

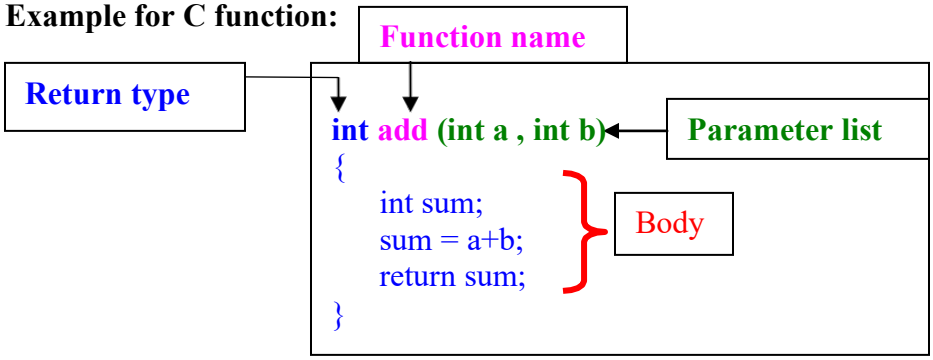
A large C program is divided into basic building blocks called a function. C function contains a set of instructions enclosed by "{ }" which performs a specific operation in a C program.

C Functions can be invoked within a C program. The main() function is the function from where every C program is started to execute. Name of the function is unique in a C program.

Structure of a C function:

```
return_type function_name (list of arguments separated by comma)
{
    Body of function;
}
```

Example for C function:



Before calling and defining a function, **we have to declare the function prototype in order to inform the compiler about the function name, function parameters and return value type.**

```
#include<stdio.h>
int add (int a , int b);        <- Declaration of function
void main( ) {
  int a, b ;
int add (int a , int b)
{
    int sum;
    sum = a+b;
    return sum;
}
printf ( "a+b=%d" ,add(5,6));
}
```

Example program for C function:

```
#include<stdio.h>
// function prototype, also called function declaration
float square ( float x );
// main function, program starts from here
void main( )
{
        float m, n ;
       printf ( "\nEnter some number for finding square \n");
       scanf ( "%f", &m ) ;
       // function call
       n = square ( m ) ;
       printf ( "\nSquare of the given number %f is %f",m,n );
 }
 float square ( float x ) // function definition
{
       float p ;
       p = x * x ;
       return ( p ) ;
}
```

Output:

```
Enter some number for finding square
2
Square of the given number 2.000000 is 4.000000
```

How to call a C function?

There are two ways that a C function can be called from a program. They are,

1. call by value
2. call by reference

1. Call by value:

Arguments are passed by value while calling a function in C program.

Example program for C function (using call by value):

```c
#include <stdio.h>
void swap(int a, int b);

int main() {
    int x = 22, y = 44;
    printf("Before swap x=%d and y=%d\n", x, y);
    swap(x, y);
    printf("After swap x=%d and y=%d\n", x, y);
}

void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}
```

Output:
Before swap x=22 and y=44
After swap x=22 and y=44

1. Call by reference:

Arguments are passed by address while calling a function in C program.

Example program for C function (using call by reference):

```c
#include <stdio.h>
void swap(int *a, int *b);

int main() {
    int x = 22, y = 44;
    printf("Before swap x=%d and y=%d\n", x, y);
    swap(&x, &y);
    printf("After swap x=%d and y=%d\n", x, y);
}

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

Output:
Before swap x=22 and y=44
After swap x=44 and y=22

Exercises

1. Write a C program to receive two integers and return their multiplication aid of a function. Use the function name as **multi.**

2. Write a C program to receive two integers and return their sum of squares aid of a function. Use the function name as **sumOfSquare**.

3. Write a C program to receive the base and height of a triangle in double type values and return its area aid of a function. Use the function name as **triangleArea**.

4. Write a C program to receive any integer value and return its factorial value aid of a function. Use the function name as **fact**.

5. Write a C program to receive a number of terms and return the Fibonacci series aid of a function. Use the function name **Fibo**.

6. Write a C program to receive temperature in Celsius and return its Fahrenheit value aid of a function. Use the function name as **CelciTOFahren**.