|  |  |
| --- | --- |
| **CS118 Programming Fundamentals** | **LAB 06** Functions & Recursion |
| **NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES** | |

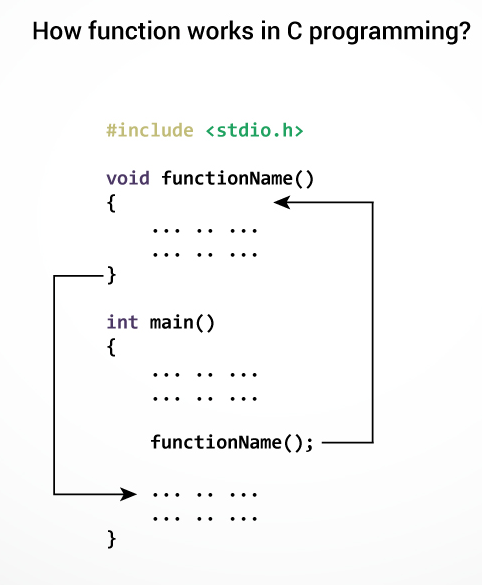
## **Function** A function is a group of statements that together perform a task. Every C program has at least one function, which is **main()**, and all the most trivial programs can define additional functions. Categories of function

**Pre-defined function:**  
 These are the functions which already have a definition in header files (.h files like stdio.h), so we just call them whenever there is a need to use them.  
**User defined function:**  
The functions that we create in a program are known as user defined functions.

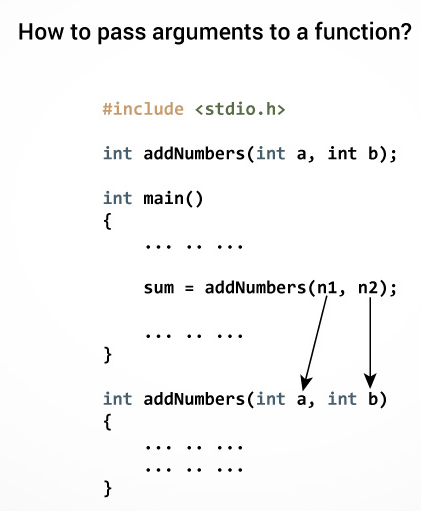
**Function Syntax**   
There are 3 aspects in each C function. They are,

* **Function declaration or prototype**: This informs compiler about the function name, function parameters and return value’s data type.
* **Function call:** This calls the actual function
* **Function definition:** This contains all the statements to be executed.

|  |  |
| --- | --- |
| C functions aspects | syntax |
| function definition | Return type function name (arguments list) { Body of function; } |
| function call | function\_name (arguments list); |
| function declaration | return\_type function\_name (argument list); |

****

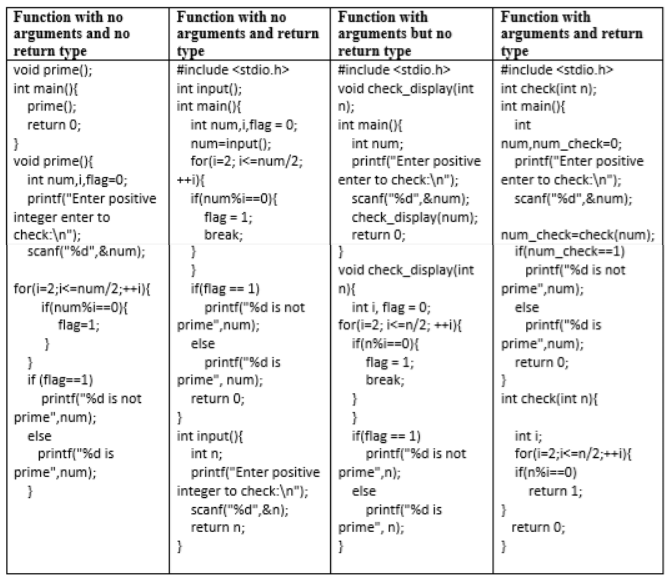
## **Passing arguments to a function**

In programming, argument refers to the variable passed to the function. The parameters a and b accepts the passed arguments in the function definition. These arguments are called formal parameters of the function.  
The type of arguments passed to a function and the formal parameters must match, otherwise the compiler throws error.  
If n1 is of char type, a also should be of char type. If n2 is of float type, variable b also should be of float type.

## **DIFFERENT NTYPES OF FUNCTIONS**

There are four types:

1. Function with no arguments and no return value
2. Function with no arguments and return value
3. Function with arguments but no return value
4. Function with arguments and return value.

**EXAMPLE:**

## **Variable Scope**

|  |  |
| --- | --- |
| Position | Type |
| [Inside a function or a block.](https://www.w3schools.in/c-tutorial/variable-scope/#local_variables) | local variables |
| [Out of all functions.](https://www.w3schools.in/c-tutorial/variable-scope/#global_variables) | Global variables |
| [In the function parameters.](https://www.w3schools.in/c-tutorial/functions/#function_argument) | Formal parameters |
|  |  |
|  |  |

#include <stdio.h>

/\* function declaration \*/

int max(int num1, int num2);

int main () {

/\* local variable definition \*/

int a = 100;

int b = 200;

int ret;

/\* calling a function to get max value \*/

ret = max(a, b);

printf( "Max value is : %d\n", ret );

return 0;

}

/\* function returning the max between two numbers \*/

int max(int num1, int num2) {

/\* local variable declaration \*/

int result;

if (num1 > num2)

result = num1;

else

result = num2;

return result;

}

### **Recursion** A function that calls itself is known as a recursive function. And, this technique is known as recursion.

### **Example: Sum of Natural Numbers Using Recursion**

#include <stdio.h>

int sum(int n);

int main()

{

int number, result;

printf("Enter a positive integer: ");

scanf("%d", &number);

result = sum(number);

printf("sum = %d", result);

return 0;

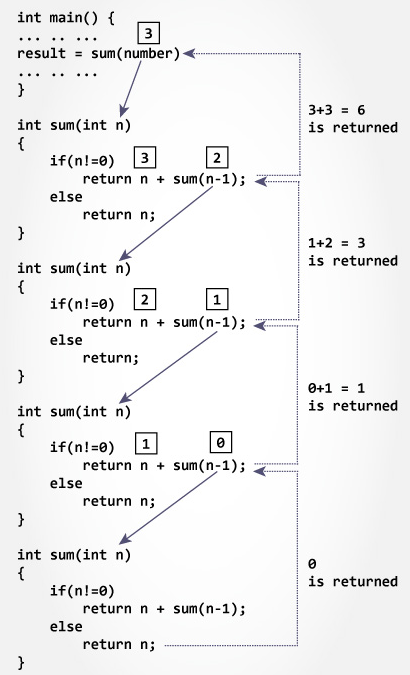
}int sum(int num)

{if (num!=0)

return num + sum(num-1); // sum() function calls itself

else

return num;}



**Lab Task**

**Question # 01:**

Write a user define function to calculate the value of an investment whose interest is compounded annually at a percentage rate of ‘r’ for ‘n’ years. The equation is given by:

totalinvestment = initialinvestment(1 + r/ 100)^n

When program was run, it produced the following sample display:

Enter the value of the initial investment: 1000

Enter the annual percentage rate of interest: 3

Enter the number of years the investment gains interest: 15

Value of final investment= 1557.97

**Question # 02:**

Write a program that asks for an employee’s salary and years of service. Write a function which changes the salary depending on the years of service. If the employee has greater than 10 years of service. He or she will receive a 10% raise. Between 5 and 10 years, the raise is 5%. Everybody else will receive a 2% raise.

Int calcbonus (int years\_service);

This function calculates and returns the employee’s bonus for the year. An employee will receive $500 for every 2 years of service.

Void printcalculation (int years\_service, float salary, int bonus);

This function will display in a clear and professional manner all of the calculated information.

**Question # 03:**

Write a function that displays a solid square of asterisks whose side is specified in integer parameter side. For example, if side is 4, the function displays:

\*\*\*\*

\*\*\*\*

\*\*\*\*

\*\*\*\*

**Question # 04:**

For the following pattern write down a function that will output the same.

\*

\*1\*

\*121\*

\*12321\*

\*1234321\*

\*123454321\*

\*1234321\*

\*12321\*

\*121\*

\*1\*

\*

Hint: first print the upper half then print lower half of the pattern.

**Question # 05:**

(a) Write a program to find Factorial of a Number Using Recursion

(b)Write a program to count the digits of a given number using recursion function.

**Question # 06:**

Write a program to GCD of two numbers using recursion function.

HINT: The GCD of 24 and 60 is 2 × 2 × 3 = 12.

**Question # 07:**

Write a program to Print Fibonacci Series using recursion function.