CC-112L

Programming Fundamentals

Laboratory 05

Introduction to Programming, Algorithms and C

Version: 1.0.0

Release Date: 09-03-2025

Department of Information Technology

University of the Punjab

Lahore, Pakistan

Contents:

Learning Objectives:

- Functions
- Structure
- Types
- Use

Resources Required:

- Desktop Computer or Laptop
- Microsoft ® Visual Studio 2022

Teachers:		
Course Instructor	Hafiz Anzar Ahmad	anzar@pucit.edu.pk
Teacher Assistants	Manahil	Bitf21m002@pucit.edu.pk

Background and Overview:

Objective:

Students will learn about functions, their purpose, syntax, types, and implementation in C++. By the end of this lab, students should be able to create and use functions effectively.

1. Introduction to Functions

What is a Function?

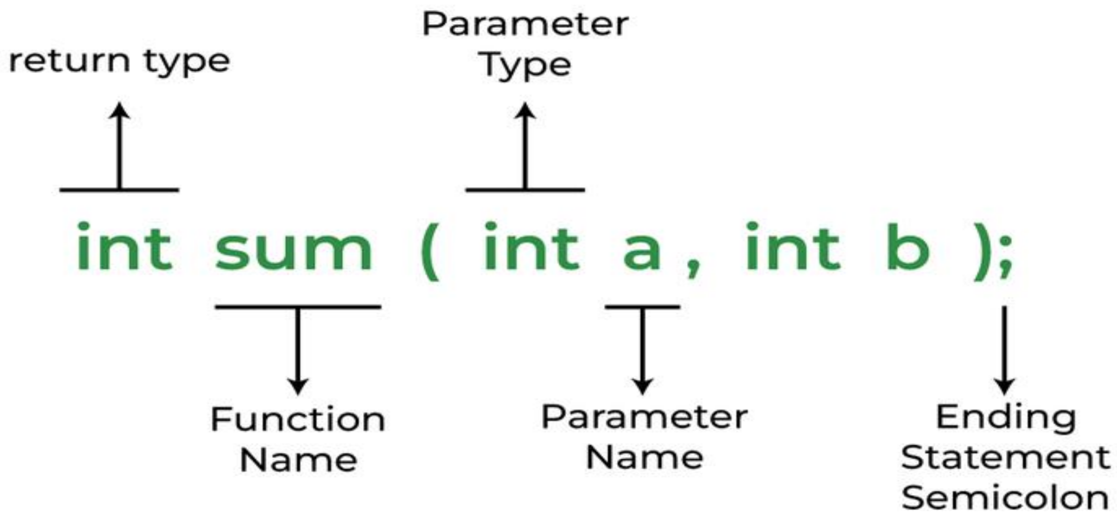
A function is a block of code that performs a specific task and can be reused multiple times. Functions help in code modularity and readability.

Why Use Functions?

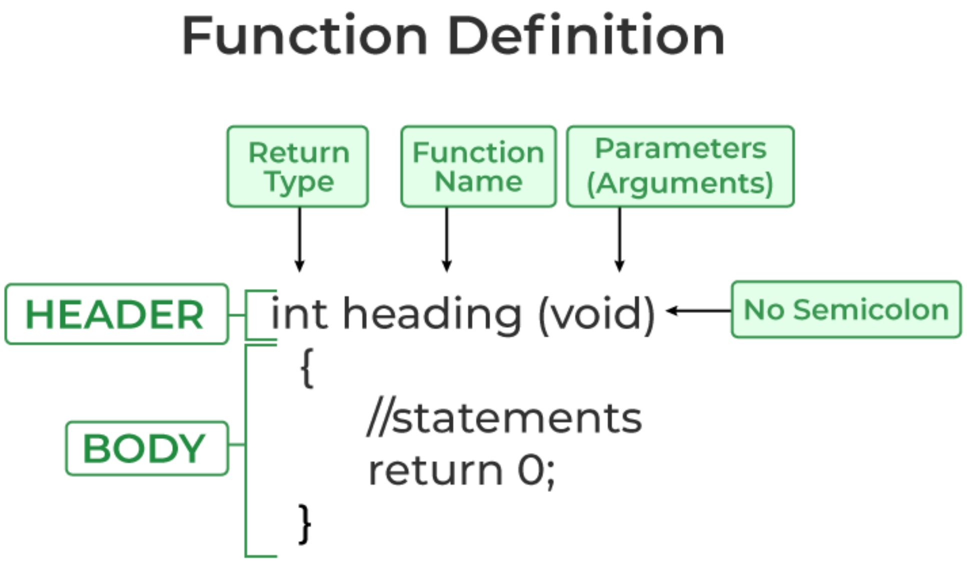
- Reduce code redundancy.
- Improve code organization.
- Enhance readability and debugging.
- Allow reusability.

A function in C consists of:

- **Function Declaration (Prototype)** – Specifies the function's name, return type, and parameters.

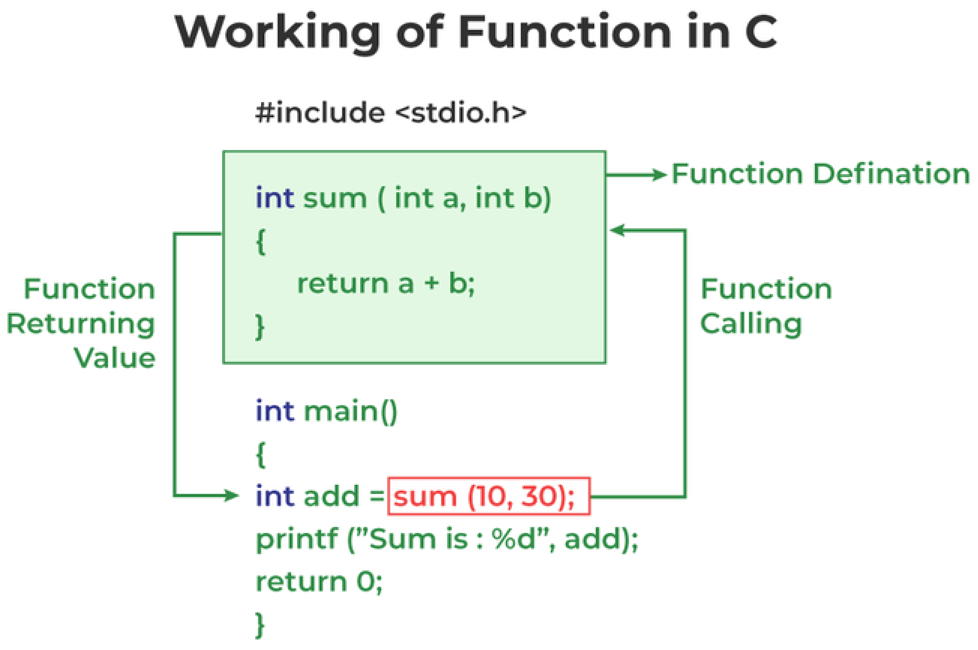


- **Function Definition (Implementation)** – Contains the actual code to execute when the function is called.



```
int heading (void)
{
    //statements
    return 0;
}
```

- **Function Call (Execution)** – Invokes the function to perform a specific task.



```
#include <stdio.h>

int sum ( int a, int b)
{
    return a + b;
}

int main()
{
int add = sum (10, 30);
printf ("Sum is : %d", add);
return 0;
}
```

2. Structure of a Function

In the following code snippet you will see its complete implementation:

```c
#include <stdio.h>

// Function Declaration (Prototype)
int add(int, int);

int main() {
    int result = add(5, 3);  // Function Call
    printf("Sum: %d\n", result);
    return 0;
}

// Function Definition (Implementation)
int add(int a, int b) {
    return a + b;
}
```

Labeled Explanation:

- **Declaration**: int add(int, int); informs the compiler about the function.
- **Definition**: int add(int a, int b) { return a + b; } defines what the function does.
- **Call**: int result = add(5, 3); invokes the function and stores the return value.

3. Types of Functions

1. Built-in Functions

C++ provides many built-in functions such as pow(), sqrt(), and abs(). Example:

```cpp
#include <iostream>
#include <cmath>  // Required for math functions
using namespace std;

int main() {
    cout << "Square root of 25: " << sqrt(25) << endl;
    return 0;
}
```

2. User-Defined Functions

Programmers can create their own functions. These can be categorized as:

- **Function without Parameters and without Return Value**

```
void greet() {
   printf("Hello, welcome to C programming!\n");
}
```

- **Function with Parameters and without Return Value**

```
void displayAge(int age) {
   printf("Your age is: %d\n", age);
}
```

- **Function with Parameters and with Return Value**

```
int getNumber() {

     return 10;

}
```

- **Function without Parameters but with Return Value**

```
int multiply(int a, int b) {

         return a * b;

}
```

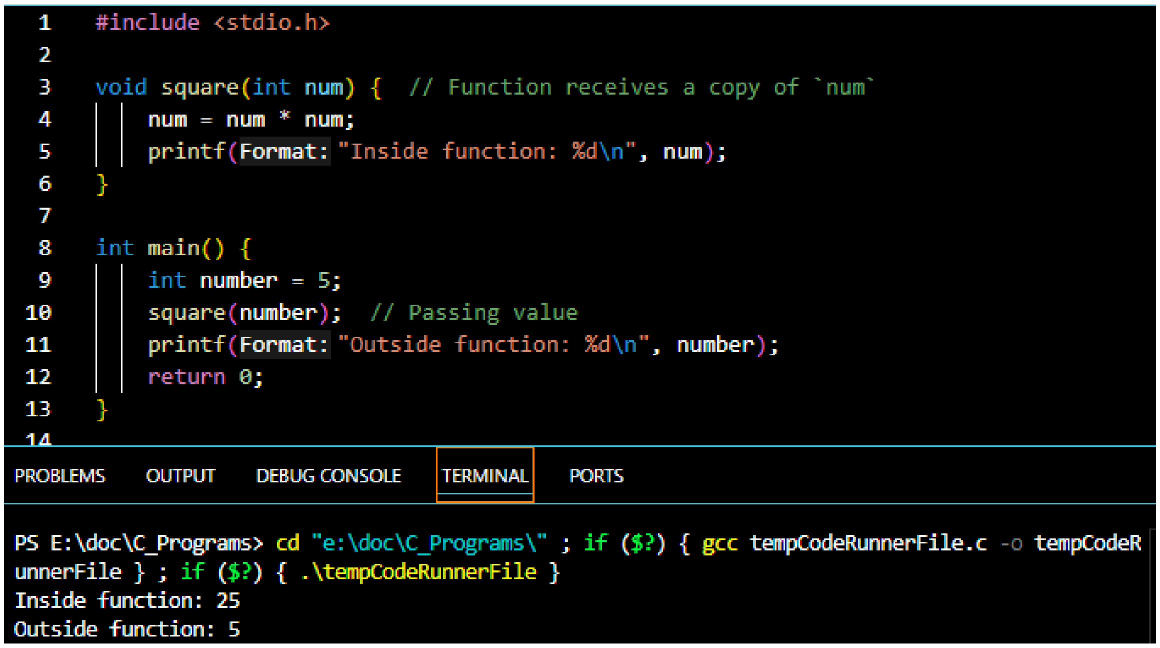
4. Function Call Methods

In C, functions can be called in two ways:

1. Call by Value

- A copy of the actual parameter is passed to the function.

- Changes inside the function do not affect the original value.


```c
#include <stdio.h>

void square(int num) {  // Function receives a copy of `num`
    num = num * num;
    printf(Format: "Inside function: %d\n", num);
}

int main() {
    int number = 5;
    square(number);  // Passing value
    printf(Format: "Outside function: %d\n", number);
    return 0;
}
```

```
PS E:\doc\C_Programs> cd "e:\doc\C_Programs\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeR
unnerFile } ; if ($?) { .\tempCodeRunnerFile }
Inside function: 25
Outside function: 5
```

Sample Output:

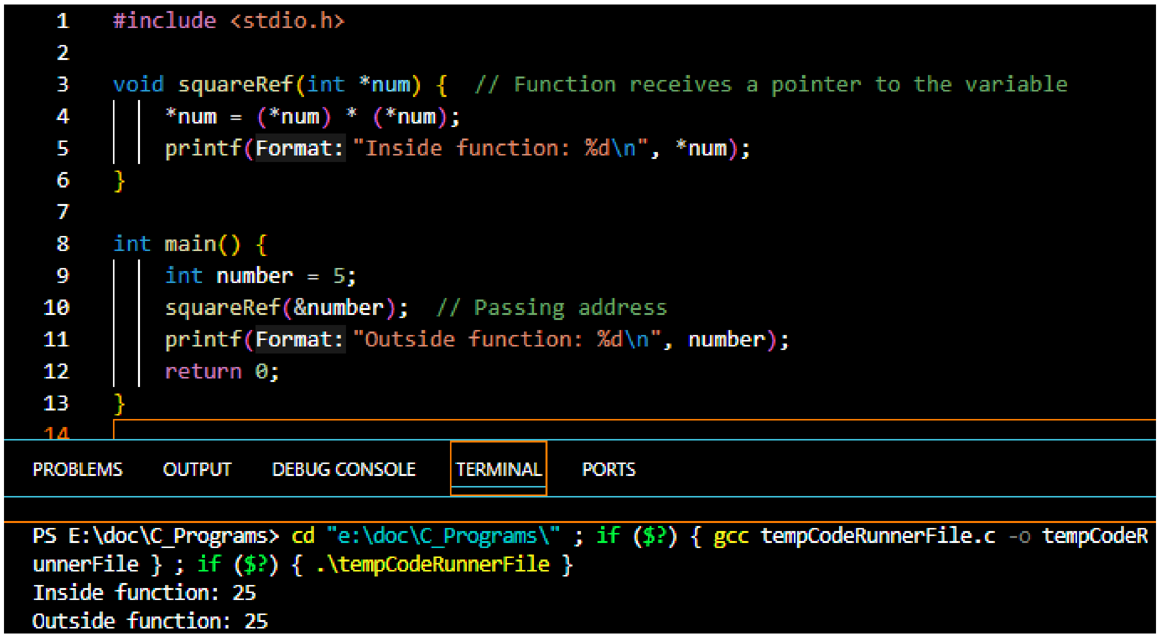
Inside function: 25

Outside function: 5

2. Call by Reference:

- Instead of passing a copy, we pass the memory address of the variable using '&'.

- The function modifies the original variable directly.


```c
#include <stdio.h>

void squareRef(int *num) {  // Function receives a pointer to the variable
    *num = (*num) * (*num);
    printf(Format: "Inside function: %d\n", *num);
}

int main() {
    int number = 5;
    squareRef(&number);  // Passing address
    printf(Format: "Outside function: %d\n", number);
    return 0;
}
```

```
PS E:\doc\C_Programs> cd "e:\doc\C_Programs\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeR
unnerFile } ; if ($?) { .\tempCodeRunnerFile }
Inside function: 25
Outside function: 25
```

There is a concept of pointers in calling a function by reference which we will discuss later in the course. So there you will get better understanding.

5. Writing Functions & Using Functions

- Identify repeated tasks in your code.
- Write a function for the repeated task.
- Call the function whenever needed.

Example: Without Functions (Repetitive Code)

```c
#include <stdio.h>
int main() {
    printf("Hello, User!\n");
    printf("Welcome to C Programming!\n");
    printf("Hello, User!\n");
    printf("Welcome to C Programming!\n");
    return 0;
}
```

Example: With Functions (Efficient Code)

```c
#include <stdio.h>
void greet() {
    printf("Hello, User!\n");
    printf("Welcome to C Programming!\n");
}

int main() {
    greet();
    greet();
    return 0;
}
```

Advantages:

- **Code reusability** – The greet() function is written once but used multiple times.
- **Better organization** – The function groups related statements together.

Pre Lab Task

Currency Converter

Objective:
You are required to implement a Currency Converter program in C using functions and a menu-driven interface. The program will allow users to:
✓ View exchange rates
✓ Convert Pakistani Rupees (PKR) to foreign currencies (USD, EUR, GBP)
✓ Convert foreign currencies to PKR

The problem will help you ro write structured progrsm using functions for better modularity and reusability.

Problem Breakdown in 5 tasks

Task 1: Display Exchange Rates [Marks 5]

Implement a function
void show_exchange_rates()   to display fixed exchange rates:
1 USD = 280 PKR
1 EUR = 300 PKR
1 GBP = 350 PKR

Example Output:

Exchange Rates:
1 USD = 280 PKR
1 EUR = 300 PKR
1 GBP = 350 PKR

Task 2: Implement a Function for Country Selection [Marks 5]

Create a function float get_exchange_rate(int option) that takes a number (1-3) and returns the exchange rate of that currency.

Bonus:
If the user enters an invalid number, return -1 to indicate an error.

To check that the function is working fine simply write a function call to it in main.
For Example
If User selects 2
Then function call, get_exchange_rate(2) → should return 300 which is EUR rate

Task 3: Convert PKR to Foreign Currency [Marks 5]

Implement float convert_pkr_to_foreign(float amount, float rate).
It should take the PKR amount and the selected country's exchange rate and return the converted amount.

Example Output
```
Enter amount in PKR: 5600
Select conversion:
1. USD
2. EUR
3. GBP
Choice: 1
Converted Amount: 20.00 USD
```

Task 4: Convert Foreign Currency to PKR [Marks 5]

Implement float convert_foreign_to_pkr(float amount, float rate).
It should take the foreign currency amount and its exchange rate to return the amount in PKR.

Example Output
```
Enter amount in foreign currency: 10
Select conversion:
1. USD
2. EUR
3. GBP
Choice: 3
Converted Amount: 3500 PKR
```

Task 5: Create a Loop/Menu-Driven Interface in the int main() function to see the implementation of all the functions you have created above. The program should keep running until the user chooses Exit. [Marks 10]

Sample Output:

```
 CURRENCY CONVERTER
1. Show Exchange Rates
2. Convert PKR to Foreign Currency
3. Convert Foreign Currency to PKR
4. Exit
Enter your choice: 1

Exchange Rates:
1 USD = 280 PKR
1 EUR = 300 PKR
1 GBP = 350 PKR

CURRENCY CONVERTER
1. Show Exchange Rates
2. Convert PKR to Foreign Currency
3. Convert Foreign Currency to PKR
4. Exit
Enter your choice: 2

Enter amount in PKR: 10000
Select conversion:
1. USD
2. EUR
3. GBP
Choice: 2
Converted Amount: 33.33 EUR

 CURRENCY CONVERTER
1. Show Exchange Rates
2. Convert PKR to Foreign Currency
3. Convert Foreign Currency to PKR
4. Exit
Enter your choice: 4
Exiting…
```