Function

A function is a block of code that performs a specific task.

Functions in C have a **name**, can accept **parameters**, and can **return a value**.

Key Benefits of Functions:

Modularity: Divides the program into small, manageable pieces.

Reusability: You can reuse functions in multiple parts of your program or in other programs.

Simplifies Debugging: Isolating functionality into functions makes debugging easier.

Improves Code Readability: Breaking down tasks into functions makes your code easier to read and understand.

Structure of a C Function

```
return_type function_name(parameter_list) {

    // Function body
    // Code to perform the task

    return value;

    /* Optional: return a value if the
    return type is not void */
}
```

Return Type: Specifies what type of value the function returns (e.g., int, float, char). If the function doesn’t return any value, use void.

Function Name: The identifier used to call the function.

Parameter List: The list of inputs (or arguments) the function expects. If the function doesn’t take any inputs, use void inside the parentheses.

Function Body: Contains the actual code that performs the task.

Return Statement: Returns a value to the calling function (if the return type is not void).

Calling a Function

```
int result = add(5, 3); // Calling the function 'add'
```