8. Functions in C

• THEORY EXERCISE: What are functions in C? Explain function declaration, definition, and how to call a function. Provide examples.

- **Function Declaration**: This tells the compiler about the function's name, return type, and parameters. It is also known as a function prototype. The declaration is usually placed at the beginning of the program or in a header file.

Syntax:

```
return_type function_name(parameter_type1 parameter_name1, parameter_type2
parameter_name2, ...);
```

Example:

```
int add(int a, int b);
```

- **Function Definition**: This is where the actual body of the function is written. It includes the code that will be executed when the function is called.

Syntax:

```
return_type function_name(parameter_type1 parameter_name1, parameter_type2
parameter_name2, ...) {

    // function body

}
```

Example:

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

    return a + b;

}
```

- **Function Call**: This is how you invoke or execute the function. You can call a function by using its name followed by parentheses containing any required arguments.

Syntax:

```
function_name(argument1, argument2, ...);
```

Example:

```
int result = add(5, 3);
```

1. **Function Declaration**: **int add(int a, int b);** informs the compiler that there is a function named **add** that takes two integers as parameters and returns an integer.

2. **Function Definition**: The function **add** is defined after the **main** function. It takes two integers, adds them, and returns the result.
3. **Function Call**: Inside the **main** function, **add(num1, num2);** calls the **add** function with **num1** and **num2** as arguments. The result is stored in the variable **result**, which is then printed.