C++ 7_1

7.1 Function Basics

A function is a group of statements that together perform a task. All C programs made up of one or more functions. There must be one and only one main function.

You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division usually is so each function performs a specific task.

A program will be executing statements sequentially inside one function when it encounters a function call.

A function call is an expression that tells the CPU to interrupt the current function and execute another function.

The CPU 'puts a bookmark' at the current point of execution, and then calls (executes) the function named in the function call. When the called function terminates, the CPU goes back to the point it bookmarked, and resumes execution.

Function definition

```
return_type function_name( parameter list )
{
//statements
}
```

- The return_type is the data type of the value the function returns. Some functions perform the desired operations without returning a value.

- function name is the identifier by which the function can be called.

- A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.

- statements is the function's body. It is a block of statements surrounded by braces { } that specify what the function actually does.

Example

```
/* Function returning addition of 2 integers */
int add(int a, int b)
{
int total ;
total= a+b;
return total;
}
```