Functions – Task which is repetitive in nature but needs to be executed at different intervals of time can be isolated and assigned to a function.

The difference between a loop and a function is that in case of a loop, the task is executed in a continuous manner & once the loop terminates, there is no way by which that task can be executed again. However that is not the case with a function. A function in it’s lifetime can be called “n” no. of times as and when required.

Code without function Code with function

3b

3a

2b

2a

1a

1b

int main() // starting point of program

{

statement 1

statement 2

statement 3

fn(); // function call

statement 4

statement 5

fn();

statement 6

fn();

statement 7 // end of main()

}

// function definition

fn()

{

statement a

statement b

statement c

statement d

}

int main() // starting point of program

{

statement 1

statement 2

statement 3

statement a

statement b

statement c

statement d

statement 4

statement 5

statement a

statement b

statement c

statement d

statement 6

statement a

statement b

statement c

statement d

statement 7 // end of main()

}

Once a call is given to a function from main(), activities in main() are suspended temporarily & the control of the program is passed to the function. After all the statements in the function are executed, the control of the program returns back to main() & executes the statements below function call. Here main() is known as “Calling function” & fn() is “Called function”.

Advantages of using functions

1. Avoids redundancy/repetitiveness of code & thus decreases the compilation time.
2. As redundancy is avoided, code becomes more compact & hence easy to maintain.
3. It is a good idea to separate the code which typically performs logically a different task and assign it to a function, to reduce the complexity of code.
4. Division of labour will decrease the overall development time and increased productivity i.e. different programmers/developers can be assigned the job of writing different functions which perform different tasks & these functions can be made a part of different header files which can be later included/imported in required program/application.

There are 2 types of functions

1. Library functions like strlen(), strcmp() [string.h] , printf(), scanf(), [stdio.h] etc. i.e. they are a part of any std C compiler (GCC, Dev C++, Turbo C, Eclipse, Borland C)
2. User Defined functions – Application specific

There are 2 syntax to call a function

1. fn\_name(argument-list);
2. var\_name = fn\_name(argument-list);

The 1st type of functions are called as “void” functions. “void” is the return type of a function. These functions do not return any value/result back to calling function.

The 2nd type of functions are called as “non-void” functions & their return type may be any data type like int, char, float, double, etc i.e. the datatype of the result which they return to calling function.