FUNCTION DEFINITION

It is also known as function implementation. When the function is defined, space is allocated for that function in memory.

Syntax

```
returntype functionname (parameter list)
{
statements;
return (value);
}
```

Example

```
int abc(int, int, int)  // Function declaration
void main()
{
        int x,y,z;
        abc(x,y,z) // Function Call
        …
        …
}
int abc(int i, int j, int k)  // Function definition
{
        …….
        ….
        return (value);
}
```

Every function definition consists of 2 parts:

a) Header of the function
b) Body of the function

a) Header of the function

The header of the function is not terminated with a semicolon. ***The return type and the number & types of parameters must be same*** in both function header & function declaration.

Syntax:

returntype functionname (parameter list)

Where,

- Return type – data type of return value. It can be int, float, double, char, void etc.
- Function name – name of the function
- Parameter type list –It is a comma separated list of parameter types.

b) Body of the function

- It consists of a set of statements enclosed within curly braces.
- The return statement is used to return the result of the called function to the calling function.

Program:

```
#include<stdio.h>
#include<conio.h>
float circlearea(int);              //function prototype
void main()
{
        int r;
        float area;
        printf("Enter the radius \n");
        scanf("%d",&r);
        area=circlearea(r); //function call
        printf(“Area of circle =%f\n”, area);
        getch();
    }
    float circlearea(int r1)
    {
        return 3.14 * r1 * r1;          //function definition
    }
```

Output:

```
    Enter the radius
    2
    Area of circle = 12.000
```