

First string: database second string: database

Now first string is: database-one

### Lecture Note: 14

## FUNCTION

A function is a self contained block of codes or sub programs with a set of statements that perform some specific task or coherent task when it is called.

It is something like to hiring a person to do some specific task like, every six months servicing a bike and hand over to it.

Any 'C' program contain at least one function i.e main().

There are basically two types of function those are

### **1. Library function**

### **2. User defined function**

The user defined functions defined by the user according to its requirement

System defined function can't be modified, it can only read and can be used.

These function are supplied with every C compiler

Source of these library function are pre compiled and only object code get used by the user by linking to the code by linker

**Here in system defined function description:**

**Function definition** : predefined, precompiled, stored in the library

**Function declaration** : In header file with or function prototype.

**Function call** : By the programmer

### **User defined function**

Syntax:-

Return type      name of function (type 1 arg 1, type2 arg2, type3 arg3)

Return type      function name      argument list of the above syntax

So when user gets his own function three thing he has to know, these are.

### **Function declaration**

### **Function definition**

### **Function call**

These three things are represented like

```
int function(int, int, int);    /*function declaration*/  
  
main()    /* calling function*/  
{  
function(arg1,arg2,arg3);  
}  
  
int function(type 1 arg 1,type2 arg2,type3, arg3) /*function definition*/  
{  
Local variable declaration;  
Statement;  
Return value;  
}
```

### **Function declaration:-**

Function declaration is also known as function prototype. It inform the compiler about three thing, those are name of the function, number and type of argument received by the function and the type of value returned by the function.

While declaring the name of the argument is optional and the function prototype always terminated by the semicolon.

### **Function definition:-**

Function definition consists of the whole description and code of the function.

It tells about what function is doing what are its inputs and what are its out put

It consists of two parts function header and function body

Syntax:-

```
return type function(type 1 arg1, type2 arg2, type3 arg3) /*function header*/
{
    Local variable declaration;
    Statement 1;
    Statement 2;
    Return value
}
```

The return type denotes the type of the value that function will return and it is optional and if it is omitted, it is assumed to be int by default. The body of the function is the compound statements or block which consists of local variable declaration statement and optional return statement.

The local variable declared inside a function is local to that function only. It can't be used anywhere in the program and its existence is only within this function.

The arguments of the function **definition** are known as **formal arguments**.

## **Function Call**

When the function get called by the calling function then that is called, function call. The compiler execute these functions when the semicolon is followed by the function name.

Example:-

```
function(arg1,arg2,arg3);
```

The argument that are used inside the function call are called **actual argument**

Ex:-

```
int S=sum(a, b);           //actual arguments
```

## **Actual argument**

The arguments which are mentioned or used inside the function call is knows as actual argument and these are the original values and copy of these are actually sent to the called function

It can be written as constant, expression or any function call like

```
Function (x);
```

```
Function (20, 30);
```

```
Function (a*b, c*d);
```

```
Function(2,3,sum(a, b));
```

## **Formal Arguments**

The arguments which are mentioned in function definition are called formal arguments or dummy arguments.

These arguments are used to just hold the copied of the values that are sent by the calling function through the function call.

These arguments are like other local variables which are created when the function call starts and destroyed when the function ends.

The basic difference between the formal argument and the actual argument are

1) The formal argument are declared inside the parenthesis where as the local variable declared at the beginning of the function block.

2). The **formal argument** are automatically initialized when the copy of actual arguments are passed while other local variable are assigned values through the statements.

Order number and type of actual arguments in the function call should be match with the order number and type of the formal arguments.

## **Return type**

It is used to return value to the calling function. It can be used in two way as

```
return
```

```
Or return(expression);
```

```
Ex:- return (a);
```

```
return (a*b);
```

```
return (a*b+c);
```

Here the 1<sup>st</sup> return statement used to terminate the function without returning any value

```
Ex:- /*summation of two values*/
```

```
int sum (int a1, int a2);
```

```
main()
```

```

{
int a,b;
printf("enter two no");
scanf("%d%d",&a,&b);
int S=sum(a,b);
printf("summation is = %d",s);
}
int sum(intx1,int y1)
{
int z=x1+y1;
Return z;
}

```

### **Advantage of function**

By using function large and difficult program can be divided in to sub programs and solved. When we want to perform some task repeatedly or some code is to be used more than once at different place in the program, then function avoids this repetition or rewritten over and over.

Due to reducing size, modular function it is easy to modify and test

### **Notes:-**

C program is a collection of one or more function.

A function is get called when function is followed by the semicolon.

A function is defined when a function name followed by a pair of curly braces

Any function can be called by another function even main() can be called by other function.

```
main()
{
function1()
}
function1()
{
Statement;
function2;
}
function 2()
{

}
```

So every function in a program must be called directly or indirectly by the main() function. A function can be called any number of times.

A function can call itself again and again and this process is called **recursion**.

A function can be called from other function **but** a function can't be defined in another function

### **Lecture Note: 15**

#### **Category of Function based on argument and return type**

##### **i) Function with no argument & no return value**