

FUNCTIONS

- Concept of function, using functions, call by value and call by reference mechanism to working with functions-example programs, passing arrays to functions, scope and extent, storage classes, recursion.

Categories of the functions

1. Function with no parameters and no return values
2. Function with no parameters and return values.
3. Function with parameters and no return values
4. Function with parameters and return values

❖ Function with no parameters and no return values

In this category, there is no data transfer between the calling function and called function. So, calling function cannot send values and the called function cannot receive the data. Consider the following program

Ex:

```
#include<stdio.h>

void add();
void main()
{
    add();          /* Function Call */
}

void add()          /* Function Header */
{
    int a,b,c;
    printf("enter values for a and b:");
    scanf("%d %d",&a,&b);
    c = a+b;
    printf("sum is:%d",c);
}
```

In the above program

- In The calling function, when the function add() is called , no arguments are passed to the function add(). So, no parameters are defined in the function header.
- When the function add() is called , the values for a and b are read, they are added and the resulted is printed on the monitor.

❖ Function with parameters and no return values

In this category, there is data transfer from the calling function to the called function using parameters. But there is no data transfer from called function to the calling function.

Ex:

```
#include<stdio.h>
void add(int m, int n);
void main()
{
    int a,b;
    printf("enter values for a and b:");
    scanf("%d %d",&a,&b);
    add(a,b); /* Function Call */
}

void add (int m,int n) /* Function Header */
{
    int c;
    c = m+n;
    printf("sum is:%d",c);
}
```

In the above program

- In the calling function, when the function add() is called, two arguments a and b are passed to the function add(). So, two parameters m and n are defined in function header.
- The values of actual parameters a and b are copied into formal parameters m and n.
- The value of m and n are added and result stored in c is displayed on the screen

❖ Function with no parameters and return values

In this category there is no data transfer from the calling function to the called function. But, there is data transfer from called function to the calling function. When the function returns a value, the calling function receives one value from the called function.

Ex:

```
#include<stdio.h>

int add();

void main()
{
    int result;

    result=add(); /* Function Call */

    printf("sum is:%d",result);
}

int add() /* Function Header */
{
    int a,b,c;
    printf("enter values for a and b:");
    scanf("%d %d",&a,&b);
    c = a+b;
    return c;
}
```

In the above program

- In the calling function, when the function add() is called, no arguments are passed to the function add(). So, no parameters are defined in the function header.
- When the control is transferred to the called function, the two values are read, they are added and the result is stored in c.
- When a return statement is executed in the function, the function is terminated immediately and control goes to the calling function.
- The function call is replaced by the value returned by the return statement and this value is copied into result in function main.

❖ Function with parameters and return values

In this category, there is data transfer between the calling function and called function. When parameters are passed, the called function can receive the values from the calling function. When the function returns a value, the calling function can receive a value from the called function.

Ex:

```
#include<stdio.h>
int add(int a, int b);
void main()
{
    int a,b,result;
    printf("enter values for a and b:");
    scanf("%d %d",&a,&b);
    result=add(a,b); /* Function Call */
    printf("sum is:%d",result);
}
int add(int m,int n) /* Function Header */
{
    int c;
    c = m+n;
    return c;
}
```

In the above program

- In the calling function, when the function add() is called, two arguments a and b are passed to the function add(). So, two parameters m and n are defined in the function header.
- The values of actual parameters a and b are copied into formal parameters m and n.
- The values of m and n are added and result is stored in c.
- When a return statement is executed in the function, the function is terminated immediately and control goes to the calling function.
- The function call is replaced by the value returned by the return statement and this value is copied into result in function main