

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.

Storage Classes

A storage class defines the scope (visibility) and life time of variables and/or functions within a C Program.

There are following storage classes which can be used in a C Program

- auto
- register
- static
- extern

- **auto - Storage Class:** auto is the default storage class for all local variables

Example:

```
void main()
{
    int count;
    auto int count;
}
```

The example above defines two variables with the same storage class. auto can only be used within functions i.e. local variables.

- **register - Storage Class:** register is used to define local variables that should be stored in a register instead of Primary memory. This means that the variable has a maximum size equal to the register size (usually one word).

Example:

```
void main()
{
    register int count;
}
```

Register should only be used for variables that require quick access - such as loops.

- **Static-Storage Class:** The variables that are declared using the keyword `static` are called static variables. The static variables can be declared outside the function and inside the function. They have the characteristics of both local and global variables.

Static can also be defined within a function. If this is done the variable is initialized at run time but is not reinitialized when the function is called. This inside a function static variable retains its value during various calls.

/*****Example *****/

```
#include<stdio.h>
void display();
void main()
{
    int i;
    for(i=1;i<=5;i++)
        display();
    getch();
}
void display()
{
    static int n=0;
    n=n+1;
    printf("the value of n is:%d \n",n);
}
```

In the above example, we are calling `display()` function 5 times, but in the `display()` function we are initializing variable `n` to 0. Since the variable `n` is **static** it will be initialized only once.

- **Extern –Storage Class:** The variables that are declared using the keyword `extern` are called extern variables. These variables are declared before all functions in global area of the program.

Note: Location of functions

While writing the program making use of function the question arises as to where the function has to be placed in the program. Generally function placement in the program can be done in three possible ways.

1. Function to be placed first followed by the main program.
2. Function to be placed after the main program, but function prototyping has to be done before main program.
3. Two separate files one for each function and main program and compiled separately.

In the first case the definition of the function appears before the main program's call to the function, so that by the time compiler reaches the function call it will be knowing what to expect.

Header files and Global declarations
Function definition
Main program

In the second case the function appears after the main program. In this case the compiler priorly doesn't know about the function exact when it encounters the function call in the main program. To overcome this we are specifying the function prototyping.

Header files and Global declarations
Function prototyping
Main program
Function definition

In the last case we maintain function in a separate file and main program in a separate file, compile them separately and a link will be provided between the two for communication. This is the way how exactly the library functions work.

File 1
Header file and function prototyping
Main program

File 2
Function definition