Tutorial Link https://codequotient.com/tutorials/Storage Classes/59fdef58e63d6b7fd5dec066

TUTORIAL

Storage Classes

Chapter

The main properties of storage classes can be summarized as below:
-

Storage Class	Storage Place	Scope	Default Value	Life Time
auto	memory	local	garbage	Till the control is in the block.
register	registers	local	garbage	Till the control is in the block.
extern	memory	global	ZERO (0)	Till the end of the main program
static	memory	local	ZERO (0)	Till the end of the main program

Guidelines for storage classes

To improve the speed of execution of the program and to carefully use the memory space by the variables, following points should be kept in mind while using storage classes: -

- Use static storage class only when we want the value of a variable to remain same every time we call it using different function calls.
- Use register storage class only for those variables that are often used in our program. CPU registers are limited and thus should be

used carefully. For example, for variables used in calculations of a loop as they will required at each loop iteration.

- Use external or global storage class only for those variables that are being used by almost all the functions in the program. For example, the value of a pi (in mathematics) which required by all modules of math library etc.
- If we do not have the purpose of any of the above mentioned storage classes, then we should use the automatic storage class. Most of the times, we use automatic storage class.

A program may contain as many variables of as many storage classes i.e.: -

C

```c
#include <stdio.h>
static int num=5;
int main() {
        register int num1=20;
        register int num2;
        int num3 = 7;
        char ch1 = 'A';
        printf("num = %d \nnum1=%d \nnum2=%d \nnum3=%d \nch1=%c\n",num,num1,num2,num3,ch1);
  return 0;
}

```

C++

```cpp
#include<iostream>
#include<cstdio>
#include<cmath>
using namespace std;

static int num=5;
int main() {
        register int num1=20;
        register int num2;
        int num3 = 7;
        char ch1 = 'A';
        cout<<"num = "<<num<<"\nnum1="<<num1<<"\nnum2="<<num2<<"\nnum3="<<num3<<"\nch1="<<ch1;
```

```
13    return 0;
14 }
15
```