Storage Classes in C

We use the storage class in the C language for determining the visibility, lifetime, initial value, and memory location of any given variable. These classes precede the type that they are going to modify.

Types of Storage Classes in C

There are four different types of storage classes that we use in the C language:

- **Automatic Storage Class**
- **External Storage Class**
- **Static Storage Class**
- **Register Storage Class**

Use of Storage Class in C

A variable given in a C program will have two of the properties : **storage class and type**. Here, type refers to any given variable's data type, while the storage class determines that very variable's lifetime, visibility, and its scope.

Automatic

- Automatic variables are allocated memory automatically at runtime.
- The visibility of the automatic variables is limited to the block in which they are defined.

  The scope of the automatic variables is limited to the block in which they are defined.
- The automatic variables are initialized to garbage by default.
- The memory assigned to automatic variables gets freed upon exiting from the block.
- The keyword used for defining automatic variables is auto.
- Every local variable is automatic in C by default.

Example

```
#include <stdio.h>
int main()
{
int a; //auto
char b;
float c;
printf("%d %c %f",a,b,c); // printing initial default value of automatic variables a, b, and c.
return 0;
}
```

Static

- The variables defined as static specifier can hold their value between the multiple function calls.
- Static local variables are visible only to the function or the block in which they are defined.
- The same static variable can be declared many times but can be assigned at only one time.
- The default initial value of the static integral variable is 0 otherwise null.
- The visibility of the static global variable is limited to the file in which it has been declared.
- The keyword used to define static variable is static.

```
#include<stdio.h>
static char c;
static int i;
static float f;
static char s[100];
void main ()
{
printf("%d %d %f %s",c,i,f); // the initial default value of c, i, and f will be printed.
}
```

Register

- The variables defined as the register are allocated the memory into the CPU registers depending upon the size of the memory remaining in the CPU.
- We cannot dereference the register variables.we cannot use &operator for the register variable.
- The access time of the register variables is faster than the automatic variables.
- The initial default value of the register local variables is 0.
- The register keyword is used for the variable which should be stored in the CPU register.
- We can store pointers in the register. register can store the address of a variable.

```
#include <stdio.h>
int main()
{
register int a; // variable a is allocated memory in the CPU register.
printf("%d",a);
}
```

External

- The external storage class is used to tell the compiler that the variable defined as extern is declared with an external linkage elsewhere in the program.
- The variables declared as extern are not allocated any memory. It is only declaration and intended to specify that the variable is declared elsewhere in the program.
- The default initial value of external integral type is 0 otherwise null.
- We can only initialize the extern variable globally.
- An external variable can be declared many times but can be initialized at only once.

```
#include <stdio.h>
int main()
{
extern int a;
printf("%d",a);
}
```

What type of variables are stored in the storage classes?

The storage classes store the local variables but can give reference to a global variable. Local variables are those variables that are declared inside any given block. These are also known as automatic variables. They only exist inside those blocks in which we declared them. On the other hand, global variables are declared outside any function.