STORAGE CLASSES

UNIT - 1

Basics:

- **A storage class defines the scope (visibility) and life-time of variables and/or functions within a C Program.**
- **Every variable in C programming has two properties: type and storage class.**
- **Type refers to the data type of a variable. And, storage class determines the scope, visibility and lifetime of a variable.**

We have 4 storage classes in c

- **auto**
- **register**
- **static**
- **extern**

auto or local variable

The auto storage class is the default storage class for all local variables.

```
{
  int mount;
  auto int month;
}
```

The variables declared inside a block are automatic or local variables.
The local variables exist only inside the block in which it is declared.

extern or global variable

The extern storage class is used to give a reference of a global variable that is visible to ALL the program files. When you use 'extern', the variable cannot be initialized however, it points the variable name at a storage location that has been previously defined.

When you have multiple files and you define a global variable or function, which will also be used in other files, then *extern* will be used in another file to provide the reference of defined variable or function.

x.c

```c
#include <stdio.h>

int count ;

extern void write_extern();

main() {

  count = 5;

  write_extern();

}
```

y.c

```c
#include <stdio.h>

extern int count;

void write_extern(void)
 {

  printf("count is %d\n", count);

 }
```

```
count is 5
```

register storage class

- **The register storage class is used to define local variables that should be stored in a register instead of RAM.**
- **the variable has a maximum size equal to the register size (usually one word)**
- **The register should only be used for variables that require quick access such as counters.**
- **It should also be noted that defining 'register' does not mean that the variable will be stored in a register. It means that it MIGHT be stored in a register depending on hardware and implementation restrictions.**

```
{

  register int  miles;

}
```

static storage class

The static storage class instructs the compiler to keep a local variable in existence during the life-time of the program instead of creating and destroying it each time it comes into and goes out of scope. Therefore, making local variables static allows them to maintain their values between function calls.

In C programming, when static is used on a global variable, it causes only one copy of that member to be shared by all the objects of its class.