```
/*
Storage Classes in C:
-A storage class defines scope, default intial value and lifetime of a
variable.
-Memory assigned to a program can be broke down in four segments:
(i).Code
(ii).Static/Global variables
(iii).Stack
(iv).Heap
-Scope:Where will this variable be availabe?
-Default Initial Value:If we don't initialize the variable explicitly, what will
be the
value inside that variable?
-Lifetime:Life of that variable.
-In C language, following storage classes are oftenly used:
1.Automatic Variables(Auto Storage Class):
~Scope:Local to the function variable they are defined in.
~Default Value: Garbage Value(A Random Value)
~Lifetime: Till the end of the function block they are defined in.
~A function defined without any storage class specification is by default an
```

automatic variable.

~'int num' and 'auto int num' are same.

2.External Variables(External Storage Class):

~They are same as global variables.

~Scope: Global to the program they are defined in.

~Default Initial Value:0

~Lifetime: These variables are declared outside any function. They are available

throughout the lifetime of the program.

~A global variable can be changed by any function in the program.

~'int num' defined outside any function will tell compiler that num is a global

variable.

~It is recommended to minimize the use of unnecessary global variables in a

program.

~Extern keyword is used to inform our C compiler that a given variable is declared somewhere else.

~Using extern will not allocate space for the variable.

3.Static Variables(Static Storage Class):

```
~Scope: Local to the variable they are defined in.
~Default Initial Value:0
~Lifetime: They are available throughout the lifetime of the program.
~A static variable remains under existence for use within the function for entire program run.
~'static int num' written inside any function will tell compiler that num is a static variable.
~It is recommended to minimize the use of unnecessary static variables in a program.
4.Register Variables(Register Storage Class):
~Scope: Local to the function they are declared in.
~Default Initial Value: Garbage value
~Lifetime: They are available till the end of the function block, in which the variable is defined.
~Register variables request the compiler to store the variable in the CPU register
instead of storing in the memory to have faster access.
~Generally this is done for the variables which are being used frequently.

*/
```

```c
#include <stdio.h>

#include "Functions.c"

extern int a;

int num;

int main()

{

printf("%d",a);

return 0;

}
```