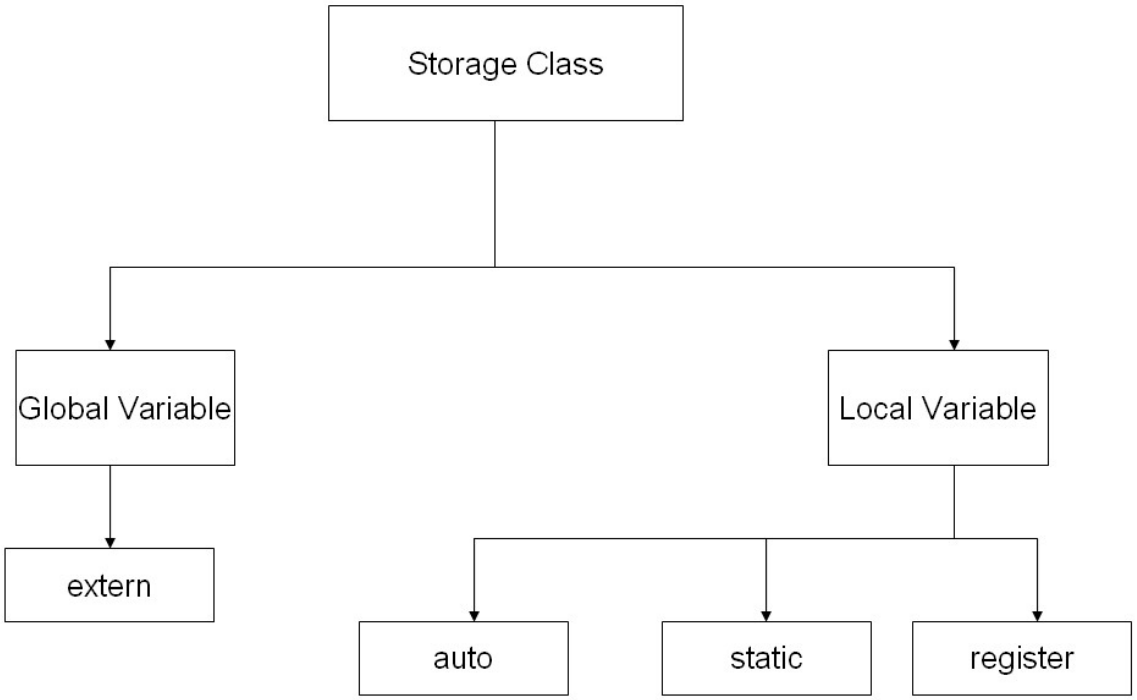
STORAGE CLASSES

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

A storage class specifies how the variables are used in the program.
The Storage class contains:

- Where the variable would be stored?
- What will be the initial value of the variable, if the initial value is not specifically assigned? (i.e. the default initial value)
- What is the scope of the variable?
- What is the life of the variable? (i.e. how long the variable would be exist?)



There are four Storage classes in c

1. Automatic Storage Class
2. Register Storage Class
3. Static Storage Class
4. External Storage Class

1. Automatic Storage Class

auto is the default storage class for all local variables.

Storage	Memory
Default storage Value	An unpredictable values, which is often called a garbage value.
Scope	Local block in which the variable is defined.
Life	Till the control remains within the block in which the variable is defined.

Eg.

```
#include<stdio.h>
#include<conio.h>
void main()
```

```
{
    auto int c=1;
    clrscr();
    {
        printf("%d",c);
    }
    printf("\n%d",c);

    getch();
}
```

2. Register Storage Class

register is used to define local variables that should be stored in a register instead of RAM. This means that the variable has a maximum size equal to the register size (usually one word) and cant have the unary '&' operator applied to it (as it does not have a memory location).

Storage	CPU Memory
Default storage Value	Garbage Value
Scope	Local block in which the variable is defined.
Life	Till the control remains within the block in which the variable is defined.

Eg.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    register int c=1;
    clrscr();
    for(c=0;c<1;c++)
        printf("\n%d",c);
    printf("\n%d",c);

    getch();
}
```

3. Static Storage Class

static is the default storage class for global variables. The two variables below (count and road) both have a static storage class.

Storage	Memory
Default storage Value	Zero
Scope	Local block in which the variable is defined.
Life	Value of variable is available between different function call.

Eg.

```
#include<stdio.h>
#include<conio.h>
extern int c=1;
void main()
```

```
{
    clrscr();
    inc();
    inc();
    dec();
    dec();
    getch();
}
inc()
{
    c=c+1;
    printf("%d\n",c);
}
dec()
{
    c=c-1;
    printf("%d\n",c);
}
```

4. External Storage Class

extern 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 initalized as all it does is point the variable name at a storage location that has been previously defined.

Storage	Memory
Default storage Value	Zero
Scope	Global
Life	As long as the program's execution doesn't come to an end

Eg.

```
#include<stdio.h>
#include<conio.h>
extern int c=1;
void main()
{
    clrscr();
    inc();
    inc();
    dec();
    dec();
    getch();
}
inc()
{
    c=c+1;
    printf("%d\n",c);
}
dec()
{
    c=c-1;
```

```
        printf("%d\n",c);
}
```

Local Variable:

Local variable are defined inside the function only. The scope of local variable is limited to only the function in which they are defined and they accessible only in the body of the same function but not outside.

```
void main()
{
        int i;
        .....
        .....
}
```

Here, variable 'i' is local function main(). When control moves outside the main(), it will no longer remain accessible, even though it us still in memory. The local variables are created when control enters into the function and are removed when function is over.

Global Variable:

The Global Variables are declared outside the function and it is accessible in more than one function.

```
int y=5;
void main()
{
        ......
        ......
}
void fun()
{
        ......
        ......
}
```

Here, the variable 'y' is declared globally with initial value 5 and shared by both the function main() and fun(). The global variables are created when program starts and removed only when program terminates. Hence, The life of global variable is same as the life of the program.

Storage Class	**Meaning**
extern	Global Variable is identified, common to all functions.
auto	Local Variable identified to the function in which it is declared.
static	It retains the same value even if the control is transferred to the function when it is called
register	It is stored in a register.