Storage Class

In

C Programming Language

Introduction

- **Basically, computer allocates space for variables in TWO ways**
  - **Memory**
  - **CPU Registers**

Scope of Variables

- **The range of code in a program over which a variable has a meaning is called as scope of the variable.**

- **Scope of a variable decides as to how it can be accessed by the different parts of the program.**

Local Scope

- A block of statement in C is surrounded by curly braces i.e { and }.
- We can declare variables inside the block, such variables called as local,can be referenced only within the body of the block.
- When the control reaches the opening brace ,these variables come into existence and get destroyed as soon as the control leaves the block through the curly brace.

Example

```
While(flag)
{
        int x;
                if(i>x)
        {
                 int  j;
                 ……….
        }
}
```

1. Scope of **x** is the while block.
2. Scope of **j** is only in the if block.
3. The variable **x** can also be referenced to in the if block because this block is completely enclosed in the while block.

The scope rules can be summarized as below

- A local variable can be accessed only within the block in which it is declared. The local variable declared in a function exists only during the execution of the function.
- Therefore these variables are also known as ***automatic*** because they are automatically created and destroyed.
- The global variables are not declared in a particular function, they are available to all the functions. Scope of a global variable is in the entire program.

Life-Time

It's the length of a time a variable store a particular value.

The ***lifetime*** of a variable is the interval of time in which storage is bound to the variable.

Why Storage Classes?

- **Where the variable would be stored?**
  - Memory or CPU Registers
- **What will be the initial value of the variable?**
  - i.e., default value (Starting value)
- **What is the scope of the variable?**
  - For which function the value of the variable would be available
- **What is the life time of a variable?**
  - How long would be variable exists

Types Storage Classes

- **There are FOUR types of storage classes**
  - Automatic Storage class (auto)
  - Register Storage class (register)
  - Static Storage class (static)
  - External Storage class (extern)

Automatic Storage Class

Storage	Memory
Default value	Garbage value
Scope	Local to the block in which the variable is defined
Life time	Till the control remains within the block in which variable is defined

Example 1

```
#include<stdio.h>
int main(){
int j;
auto float f;
printf("%d\t%f",j,f);
return 0;
}
```

Example 2

```
#include<stdio.h>
int main(){
    int a=10;
    {
        int a=20;
        printf("%d",a);
    }
    printf(" %d",a);
    return 0;
}
```

Output: 20 10

Example 3

```
#include<stdio.h>
int main(){
    {
        int a=20;
        printf("%d",a);
    }
    printf(" %d",a);  //a is not visible here
    return 0;
}
```

Output: Compilation error

Example 4

```
#include<stdio.h>
int main(){
    int i;
    for(i=0;i<4;i++){
        int a=20;
        printf("%d",a);
        a++;
    }
    return 0;
}
```

Output: 20 20 20 20

Register Storage Class

Storage	Register
Default value	Garbage value
Scope	Local to the block in which the variable is defined
Life time	Till the control remains within the block in which variable is defined

Example 1

```
#include<stdio.h>
int main(){
    register int a=10;
    int *p;
    p=&a;
    printf("%u",p);
}
```

Output: Compilation error

Example 2

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

Output: Compilation error

Static Storage Class

Storage	Memory
Default value	Zero
Scope	Local to the block in which the variable is defined
Life time	The value of the persists between different function calls (i.e., Initialization is done only once)

Example 1

```
#include<stdio.h>
static int a;
int main(){
    printf("%d",a);
    return 0;
}
```

Output: 0

Example 2

```
#include <stdio.h>
static int i=10;
int main(){
    i=25;       //Assignment statement
    printf("%d",i);
    return 0;
}
```

Output: 25

Example 3

```
#include<stdio.h>
int main(){
   {
       static int a=5;
       printf("%d",a);
   }
   //printf("%d",a);   variable a is not visible here.
   return 0;
}
```

Output: 5

Example 4

```
#include<stdio.h>
int fun()
{
  static int count = 0;
  count++;
  return count;
}

int main()
{
  printf("%d ", fun());
  printf("%d ", fun());
  return 0;
}
```

Output: 1 2

External Storage Class

Storage	Memory
Default value	Zero
Scope	Global (i.e., Throughout the program)
Life time	As long as the program's execution does not comes to end

Example 1

```
#include <stdio.h>
int i;
int main(){
    printf("%d",i);
    return 0;
}
```

Output: 0

Example 2

```
#include <stdio.h>
extern int i;    //extern variable
int main(){
    printf("%d",i);
    return 0;
}
```

Output: Compilation error, undefined symbol i.

Example 3

```
#include <stdio.h>
extern int i=10;    //extern variable
int main(){
    printf("%d",i);
    return 0;
}
```

Output: 10

Example 4

```
#include <stdio.h>
int main(){
extern int i=10; //Try to initialize extern variable locally.
    printf("%d",i);
    return 0;
}
```

Output: Compilation error: Cannot initialize extern variable.