Storage Class

DoCSE

Kathmandu University

Storage Class Specifiers

- C has four kinds of storage classes
  - Automatic
  - Register
  - Static
  - External
- Storage class is used to denote to the compiler where ***memory*** is to be allocated for variables
- Storage class tells, what will be the ***initial value*** of the variable, if the initial value is not specifically assigned.(i.e the default initial value).
- Storage class informs the compiler the ***scope*** of the variable.Scope refers to which functions the value of the variable would be available.
- Storage class informs the compiler the ***life*** of the variable. Life refers to how long would the variable exist.

Automatic Storage Class

- All variables declared within a function are called *local* variables, by default belong to the *automatic* storage class.

Storage	Memory
Default Initial Value	Garbage Value
Scope	Local to the block in which the variable is defined
Life	Till the control remains within the block in which it is defined

Example for Automatic Storage Class

```
main()
{
        auto int i,j;
        printf("%d \n %d \n",i,j);
}
```

Output

1211

876

Example for Automatic Storage Class

```
main()
{
        auto int j=1;
        {
                auto int j=2;
                {
                        auto int j=3;
                        printf("%d\n",j);
                }
                printf("%d\n",j);
        }
        printf("%d\n",j);
}
```

Output:

3

2

1

Register Storage Class

- *Register* storage class allocates memory in CPU's high speed registers.

Storage	CPU Registers
Default Initial Value	Garbage Value
Scope	Local to the block in which the variable is defined
Life	Till the control remains within the block in which it is defined

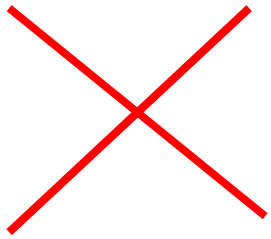
Example for Register Storage Class

```
main()
{
        register int i;
        for(i=1;i<=10;i++)
                printf("%d\n",i);
}
```

Note:

The register storage class cannot be used for all types of variables.
Example:

```
register float q;
register double s;
register long r;
```



Static Storage Class

- *Static* storage class informs the compiler that the value stored in the variables are available between function calls

Storage	Memory
Default Initial Value	Zero
Scope	Local to the block in which the variable is defined
Life	Value of the variable persists for different function calls

Example for Static Storage Class

```
main()
{
        function();
        function();
        function();
}
function()
{
        static int i=1;
        printf("%d\n",i);
        i=i+1;
}
```

Output

1

2

3

```
main()
{
        function();
        function();
        function();
}
function()
{
        auto int i=1;
        printf("%d\n",i);
        i=i+1;
}
```

Output

1

1

1

Example:2

```
int fibo(int count)
{
static int f1=1,f2=1;
int f;
f=(count<3)?1:f1+f2;
f2=f1;
f1=f;
return f;
}
```

```
main( )
{
int count,n;
printf("How many numbers:");
scanf("%d",&n);
for(count=1;count<=n;count++)
printf("count=%d f=%d",count,
fibo(count));
}
```

External Storage Class

- Variables declared outside functions have ***external*** storage class

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

Example for External Storage Class

```
int i;
main()
{
        printf("i=",i);
        increment();
        increment();
        decrement();
        decrement();
}
```

```
increment()
{
i=i+1;
printf("On incrementingi = %d\n",i);
}
```

```
decrement()
{
i=i-1;
printf("On incrementing i=%d\n",i);
}
```

Output:
i=0
On incrementing i=1
On incrementing i=2
On decrementing i=1
On decrementing i=0

Summary

- The different decision structure are if statement, if – else statement, multiple choice else if statement.
- The while loop keeps repeating an action until an associated test returns false.
- The do while loops is similar, but the test occurs after the loop body is executed.
- The for loop is frequently used, usually where the loop will be traversed a fixed number of times.
- Storage class is used to denote to the compiler where ***memory*** is to be allocated for variables
- C has four kinds of storage classes
  - Automatic
  - Register
  - Static
  - External

Thank You!