

Storage Class

Programming using C language
Dr. Nivedita Palia

Storage Class

Storage class of a variable tells us about:

Location of the variable, where it would be **stored**.

Default initial value that a variable will receive (if not assigned a value).

Scope of a variable i.e. in which function the value of variable would be available.

Lifetime of a variable i.e. how long variable exist

Storage Classes: Types

There are 4 Storage classes in C:

Automatic (auto)

Register (register)

Static (static)

External (extern)

* if storage class is not specified explicitly in the variable declaration, the compiler will assume a storage class depending on where the variable is declared

Automatic Storage Class

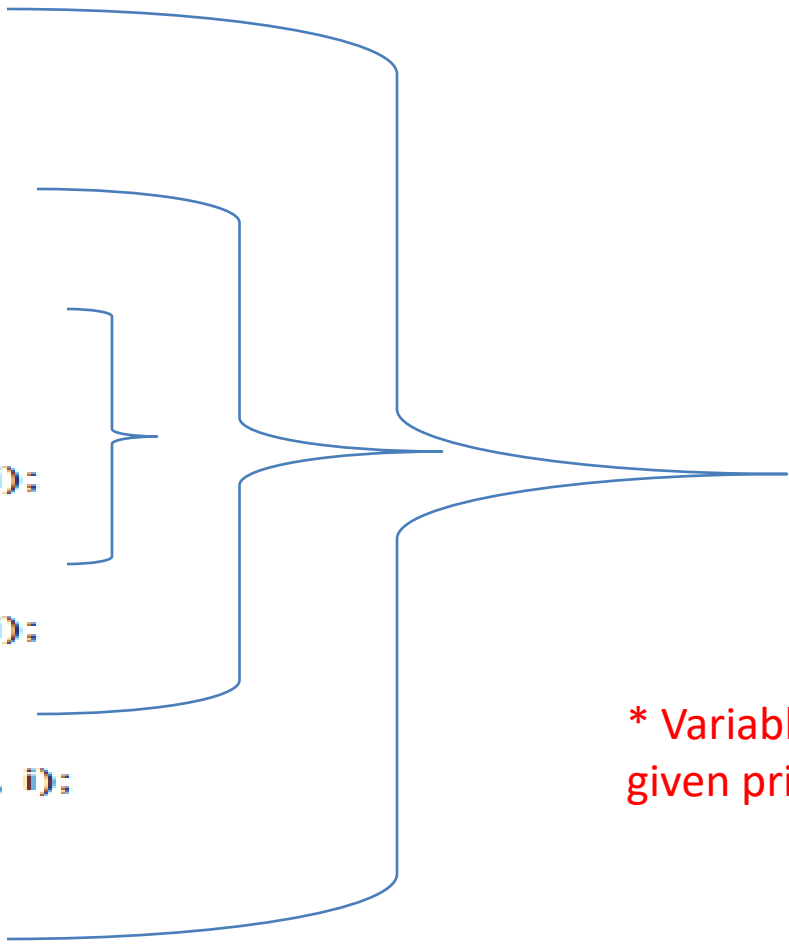
Storage Location:	Memory
Default initial value:	Unpredictable value (garbage/junk)
Scope:	Local to the block in which it is defined
Lifetime:	Till the control remains in its block

Note:- Automatic storage class is the default storage class in C.

```
void main()  
{  
    auto int i,j;  
    printf(“%d%d”,i,j);  
}
```

Auto Storage Class :Example

```
int main()  
{  
    auto int i = 1;  
    {  
        auto int i = 2;  
        {  
            auto int i = 3;  
            printf ("%d ", i);  
        }  
        printf ("%d ", i);  
    }  
    printf ("%d\n", i);  
    return 0;  
}
```



* Variable which is most local
given priority

Register Storage Class

Storage Location:

CPU Registers

Default initial value:

Unpredictable value (garbage)

Scope:

Local to the block in which it is defined

Lifetime:

Till the control remains in its block

Note:-

- 1) If CPU registers are not available, then it will behave like an automatic variable.
- 2) A value stored in CPU register can always be accessed faster
- 3) Float cannot be stored in CPU register

Static Storage Class

Storage Location: Memory

Default initial value: Zero

Scope: Local to the block in which it is defined

Lifetime: Value of variable persists between different function calls

Note: In static storage class, initialization of a variable is done only once.

Example

```
void increment();
```

```
int main()
```

```
{
```

```
increment();
```

```
increment();
```

```
increment();
```

```
return 0;
```

```
}
```

```
void increment()
```

```
{
```

```
auto int i = 1;
```

```
static int j = 1;
```

```
i = i + 1;
```

```
j = j + 1;
```

```
printf ("%d %d\n", i, j);
```

```
}
```

} Value of i initialize to 1 every time
while for j is initialize to 1 only during
first call

External Storage Class

Storage Location: Memory

Default initial value: Zero

Scope: Global

Lifetime: Till the time the program is executing

Note:- extern keyword is used to refer to an already declared/initialised variable which is external to that block

Example

```
int i;  
  
void increment();  
  
void decrement();  
  
int main()  
{  
    printf ("\ni = %d", i);  
  
    increment();  
  
    increment();  
  
    decrement();  
  
    decrement();  
  
    return 0;  
}  
  
void increment()  
{  
    i = i + 1;  
  
    printf ("on incrementing i = %d\n", i);  
}  
  
void decrement()  
{  
    i = i - 1;  
  
    printf ("on decrementing i = %d\n", i);  
}
```

END