

Scope and Lifetime

Scope (Visibility)

Lifetime (Existence)

// Global variables

int numitems = 10;

entire file
unless redeclared within
a function
From other files if declared "extern"

entire execution

static int whichitem = 0;

entire file
unless redeclared within
a function
NOT accessible from other files

entire execution

// Local variables

int fcn(.....)
{

int count;

// Within fcn only

fcn execution only
created on entry
destroyed on exit

static int totalnum = 0;

// Within fcn only

entire execution
created/initialized
on startup
persistent value
across fcn calls

}

// Class 'static'

// A member variable allocated once, but shared by all class instances

class MyClass {

static int m_NumInstances;

// Public or Private visibility

Entire execution

}

int MyClass :: m_NumInstances = 0; // Global declaration and initialization

```
// Static Data Members in a class           Simple example of use
//   Shared among all instances of the class
```

```
class point {
public:
    point(const int a=0, const int b=0);

    int HowMany() const;
```

```
private:
    int x, y;

    static int numpoints;
};
```

```
////////////////////////////////////
// Implementation File ...
```

```
int point::numpoints = 0;    // Global declaration and init
```

```
point::point(const int a, const int b)
{
    x = a;
    y = b;
    ++numpoints;
}
```

```
int point::HowMany() const
{
    return numpoints;
}
```

Example:

```
POINT  M (0, 0);
```

```
cout << M.HowMany() << endl;
```

```
POINT  N (5, 10);
```

```
POINT  P (-3, 1);
```

```
cout << P.HowMany() << endl;
```

```
cout << M.HowMany() << endl;
```

```
cout << point::numpoints << endl; // illegal- private class data member
```