Static Members in C++

Static Members

- A static class member acts as a global object among the objects of the same class
- Information hiding can be still enforced
- A static data member is not entered into programs global name space.
- A data member is made static by prefixing its declaration by *static* keyword.
- A static data member is initialized outside the class definition.
- A static data member can be a constant or a class object.

Static Members

- Static data member initialization should happen in implementation part (.cpp file)
- A member function that accesses only a static member function of a class may also be declared as static.
- A static member function does not have a “this” pointer.
- A static member function may be invoked through a class object or pointer to a class or can be accessed directly even if no class object is declared.

Static Members Example

```
// point.h
#include <iostream>
using namespace std;
class Point{
        private:
        double xcoordinate;
        double ycoordinate;
        int pointID;
        static int counter;
        public:
        Point(): xcoordinate(0), ycoordinate(0) { counter++; }
        static int getCounter() { return counter;}
};
```

Using Static Members

```
// main.cpp

int main()
{
   Point a, b;
   …
   // output is 2.
   cout << Point::getCounter();
    …
    …
}
```

```
// point.cpp

int Point::counter=0;

// other member
   functions
```