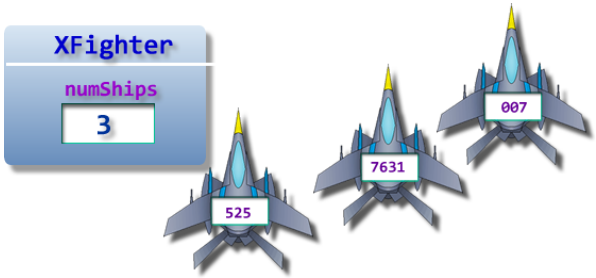
Static Data Members

Suppose you are creating a space-wars type video game, and one of your player types is an `XFighter` class. How do you easily keep track of **how many** `XFighter` ships are currently available?



That's easy: with a `static`, or **shared data member** as so:

```
class XFighter
{
public:
    XFighter(int vin) : m_VIN(vin) { numShips++; }
    ~XFighter { numShips--; }
private:
    static int numShips;
    int m_VIN;
};
```

The `static` data member `numShips` is `private` to the `XFighter` class, but there is **only one copy** of the member, **not one for each ship**, like the vehicle identification number (`m_VIN`). When an `XFighter` is constructed, the constructor increments the shared `numShips`, and, when a ship is destroyed, the destructor decrements it.

There is one wrinkle with this. In C++, you must **initialize the static member** as a global object in your `.cpp` file with something like this:

```
int XFighter::numShips = 0;
```

This **cannot** go in the header file.