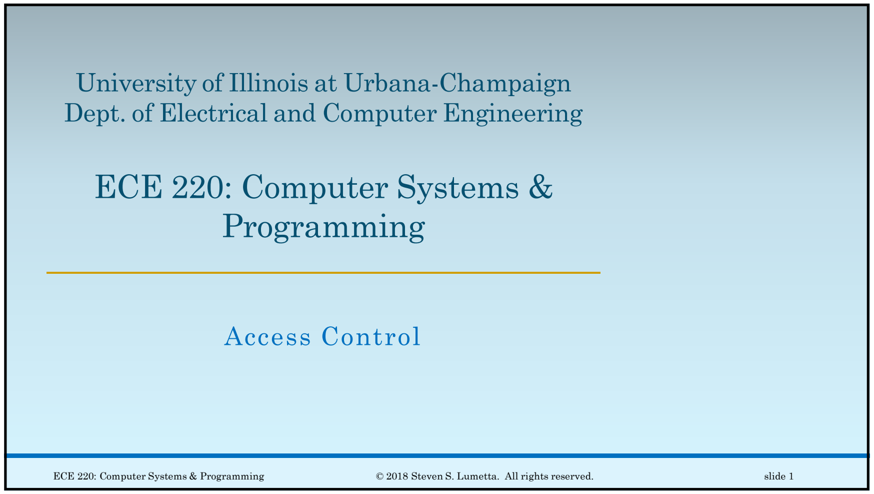
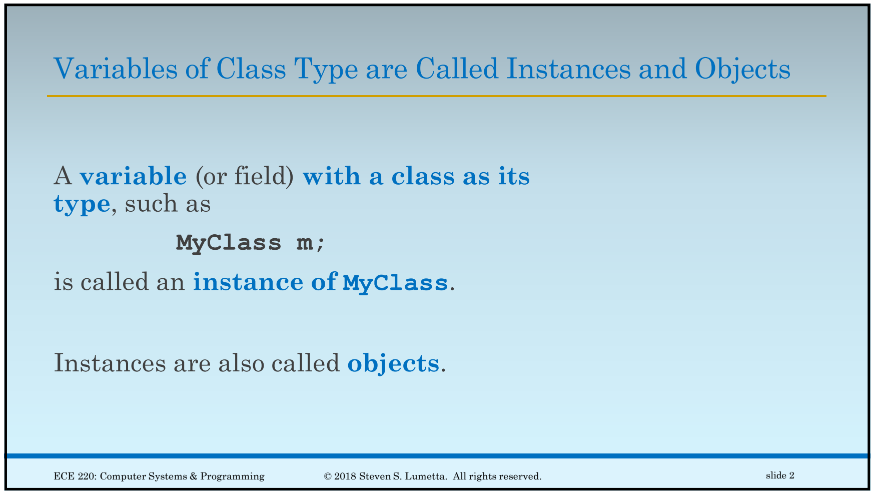


University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 220: Computer Systems & Programming

Access Control



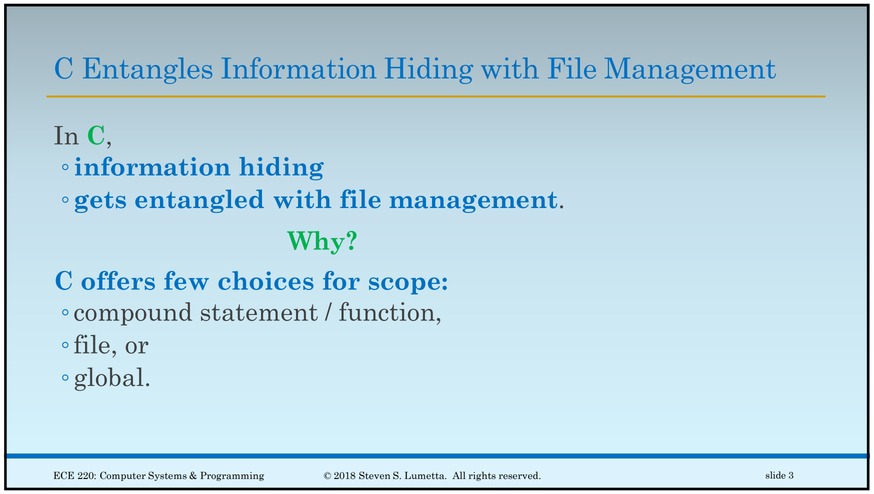
Variables of Class Type are Called Instances and Objects

A **variable** (or field) **with a class as its type**, such as

```
MyClass m;
```

is called an **instance of `MyClass`**.

Instances are also called **objects**.

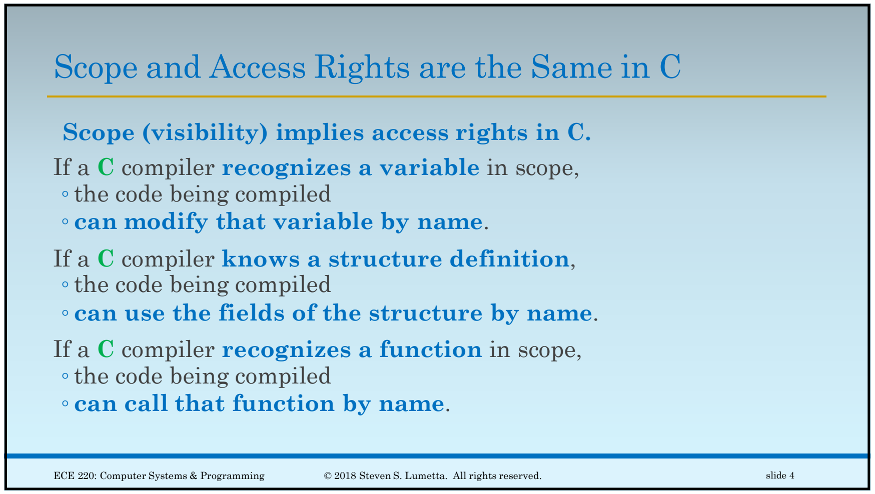


C Entangles Information Hiding with File Management

In **C**,
- **information hiding**
- **gets entangled with file management**.

Why?

C offers few choices for scope:
- compound statement / function,
- file, or
- global.



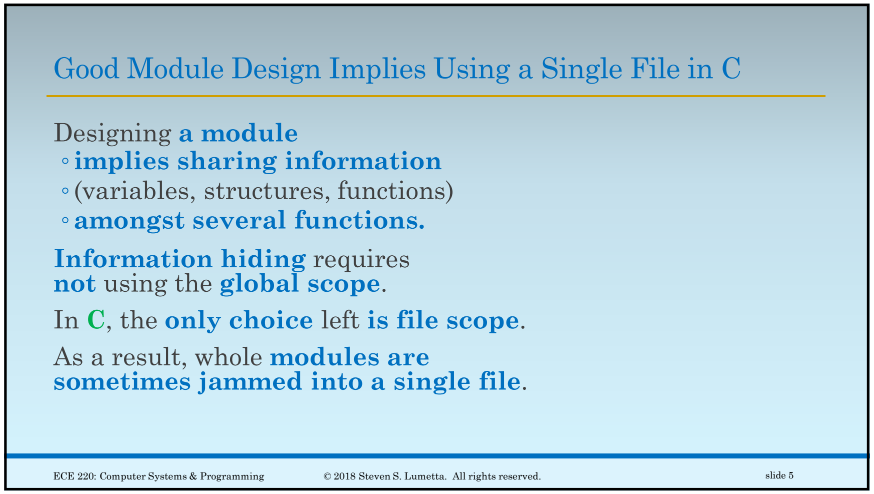
Scope and Access Rights are the Same in C

Scope (visibility) implies access rights in C.

If a **C** compiler **recognizes a variable** in scope,
- the code being compiled
- **can modify that variable by name**.

If a **C** compiler **knows a structure definition**,
- the code being compiled
- **can use the fields of the structure by name**.

If a **C** compiler **recognizes a function** in scope,
- the code being compiled
- **can call that function by name**.



Good Module Design Implies Using a Single File in C

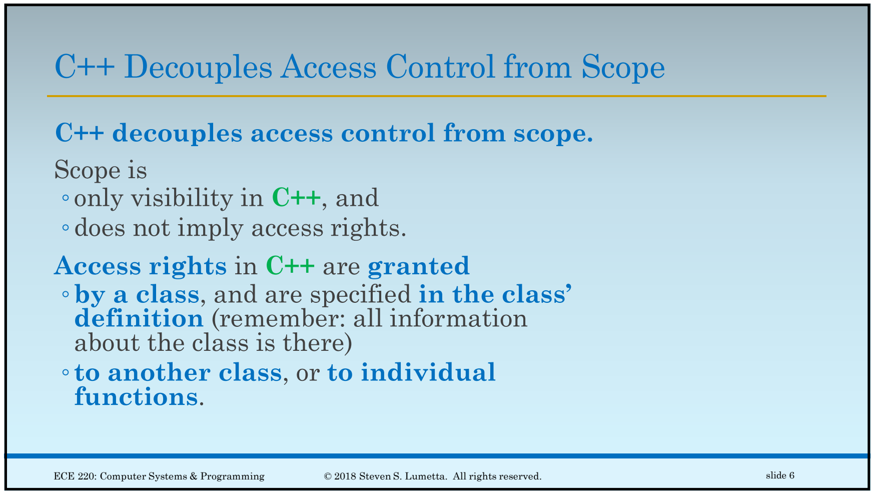
Designing **a module**

- **implies sharing information**
- (variables, structures, functions)
- **amongst several functions.**

Information hiding requires **not** using the **global scope**.

In C, the **only choice** left **is file scope**.

As a result, whole **modules are sometimes jammed into a single file**.



C++ Decouples Access Control from Scope

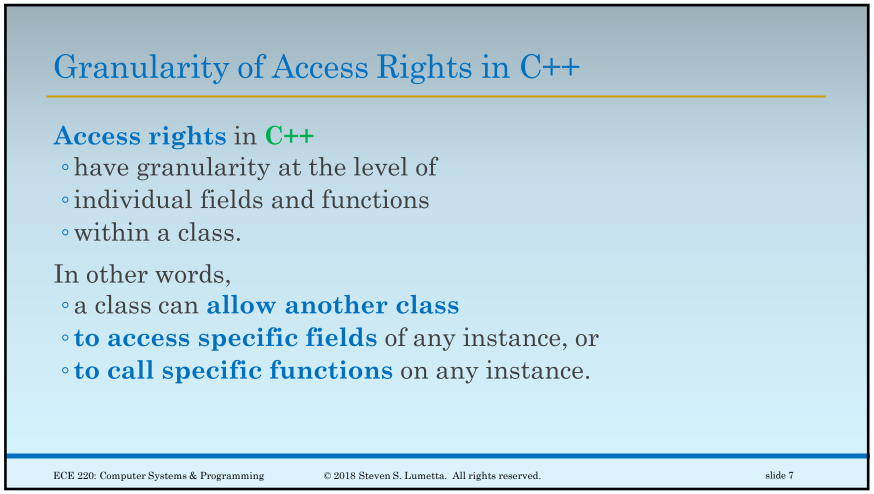
C++ decouples access control from scope.

Scope is

- only visibility in C++, and
- does not imply access rights.

Access rights in C++ are **granted**

- **by a class**, and are specified **in the class' definition** (remember: all information about the class is there)
- **to another class**, or **to individual functions**.



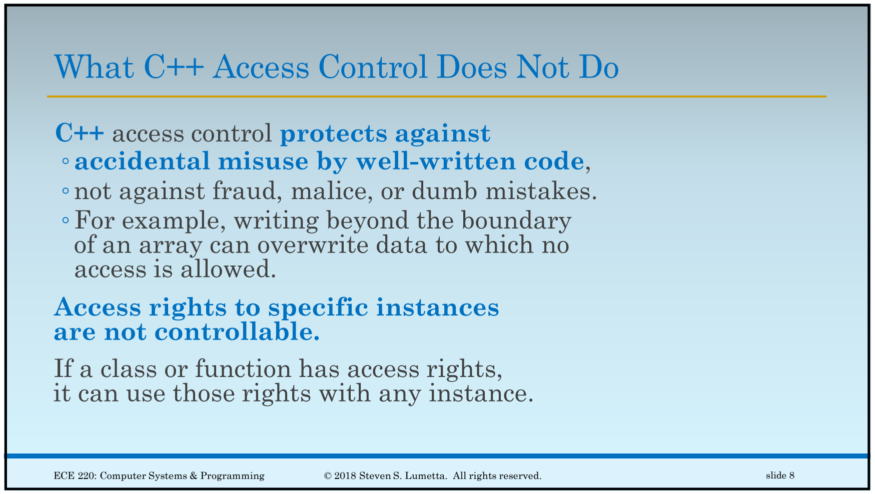
Granularity of Access Rights in C++

Access rights in C++

- have granularity at the level of
- individual fields and functions
- within a class.

In other words,

- a class can **allow another class**
- **to access specific fields** of any instance, or
- **to call specific functions** on any instance.



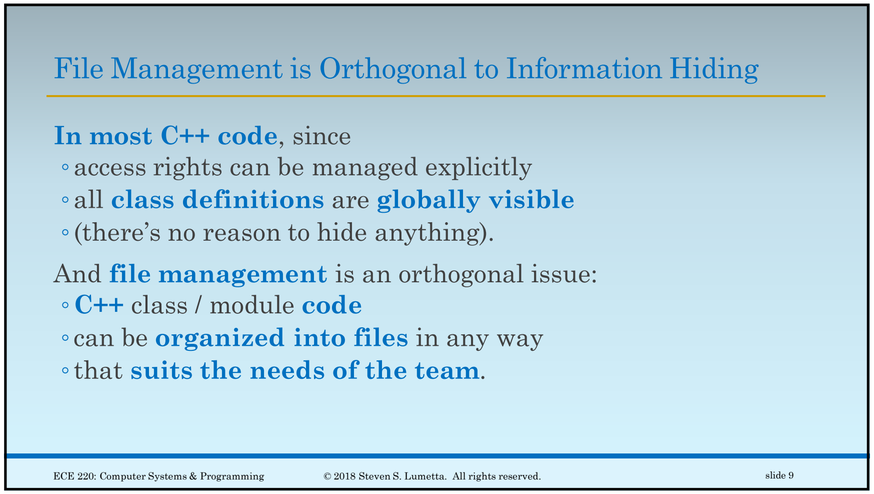
What C++ Access Control Does Not Do

C++ access control **protects against**

- **accidental misuse by well-written code**,
- not against fraud, malice, or dumb mistakes.
- For example, writing beyond the boundary of an array can overwrite data to which no access is allowed.

Access rights to specific instances are not controllable.

If a class or function has access rights, it can use those rights with any instance.

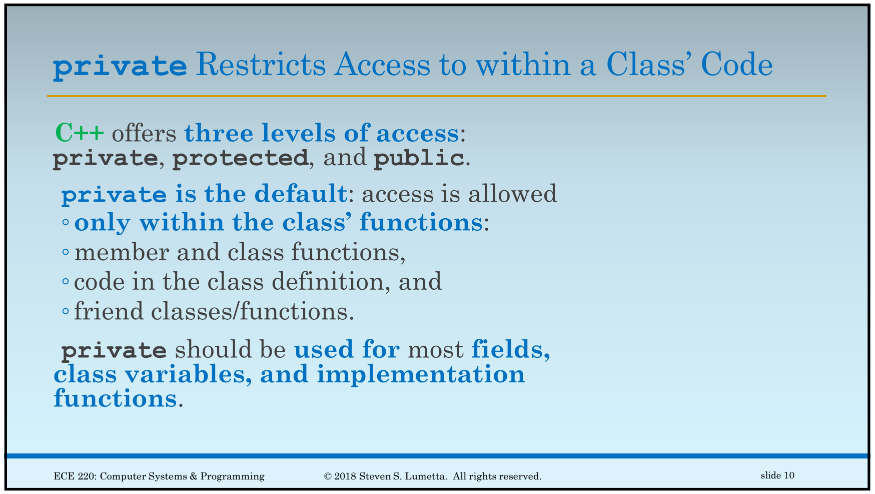


File Management is Orthogonal to Information Hiding

In most C++ code, since
- access rights can be managed explicitly
- all **class definitions** are **globally visible**
- (there's no reason to hide anything).

And **file management** is an orthogonal issue:
- **C++** class / module **code**
- can be **organized into files** in any way
- that **suits the needs of the team**.

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 9



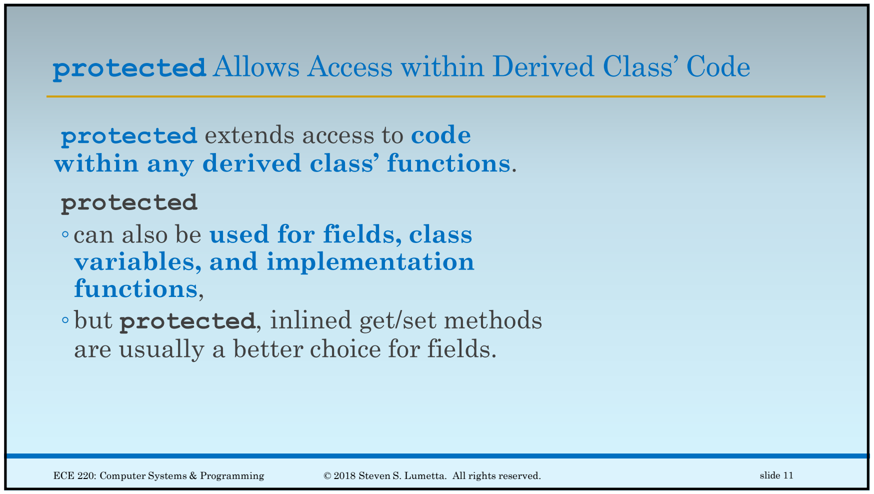
`private` Restricts Access to within a Class' Code

C++ offers **three levels of access**: **`private`**, **`protected`**, and **`public`**.

`private` is the default: access is allowed
- **only within the class' functions**:
- member and class functions,
- code in the class definition, and
- friend classes/functions.

`private` should be **used for** most **fields, class variables, and implementation functions**.

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 10

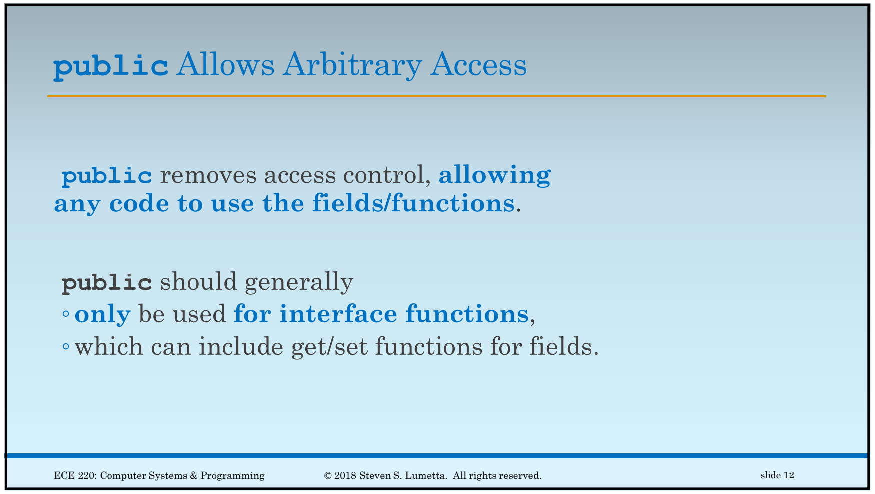


`protected` Allows Access within Derived Class' Code

`protected` extends access to **code within any derived class' functions**.

`protected`
- can also be **used for fields, class variables, and implementation functions**,
- but **`protected`**, inlined get/set methods are usually a better choice for fields.

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 11

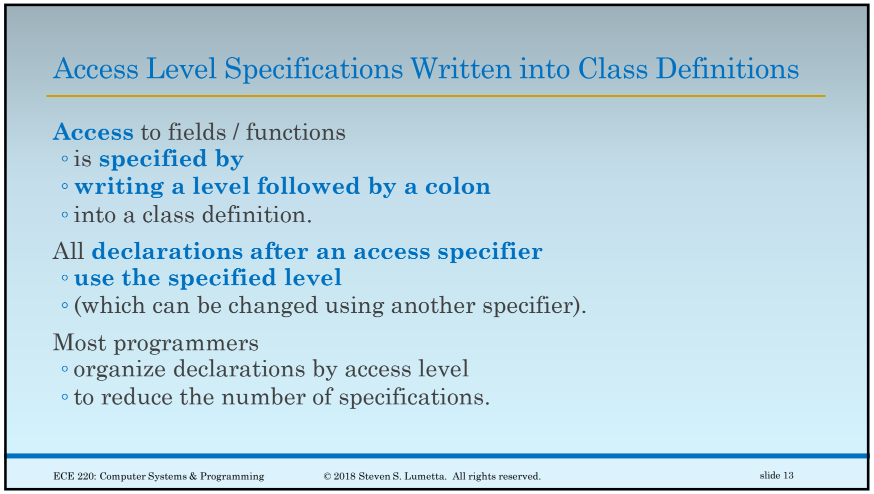


`public` Allows Arbitrary Access

`public` removes access control, **allowing any code to use the fields/functions**.

`public` should generally
- **only** be used **for interface functions**,
- which can include get/set functions for fields.

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 12

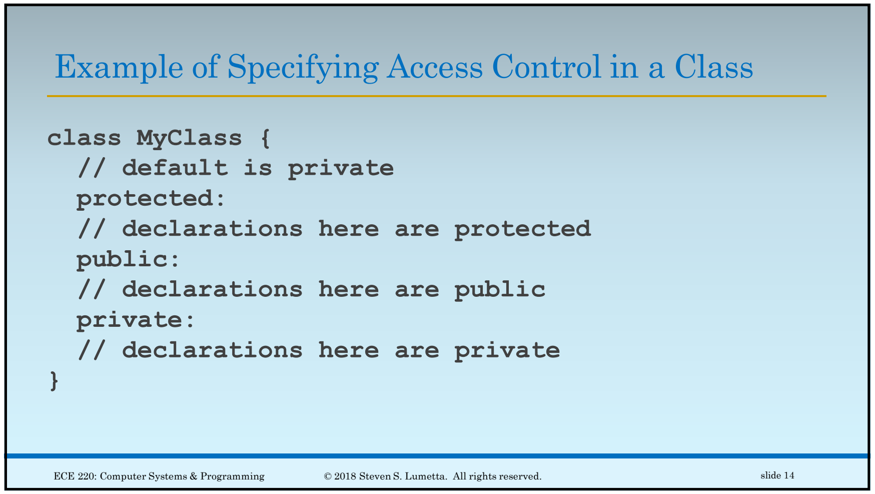


Access Level Specifications Written into Class Definitions

Access to fields / functions
- is **specified by**
- **writing a level followed by a colon**
- into a class definition.

All **declarations after an access specifier**
- **use the specified level**
- (which can be changed using another specifier).

Most programmers
- organize declarations by access level
- to reduce the number of specifications.



Example of Specifying Access Control in a Class

```
class MyClass {
  // default is private
  protected:
  // declarations here are protected
  public:
  // declarations here are public
  private:
  // declarations here are private
}
```