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**.

 © 2018 Steven S. Lumetta. All rights reserved.

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**.

 © 2018 Steven S. Lumetta. All rights reserved.

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.

 © 2018 Steven S. Lumetta. All rights reserved.

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.

 © 2018 Steven S. Lumetta. All rights reserved.