Lecture 15 CSI333

“The types in all declarations of the same object, function, class, etc., must be consistent ... the source code submitted [by the preprocessor] to the compiler and later linked together must be consistent.” (Str. 9.2.1)

- Single Header File (Strou. 9.3.1)
  - Achieves Consistancy.
  - “For larger programs ... is unworkable”
  - Not acceptable for CSI333 C++ projects.

Fundamental idea for logical organization of software: Organize it into MODULES. Each module solves a separate problem; modules will use each other’s services.
Multiple Header Files (Strou. 9.3.2)
  - Enables physical organization to reflect logical organization, plus make rebuilds efficient.
  - Each module has its own .h file that specifies its interface (what it provides).
  - The header is #include'd in the module’s .c implementation files AND in every other module that USES this module. (Guarantees consistency.)

CSI333 Software Construction Standards Header files may contain:
  - Type definitions: struct Point { int x, y; };
  - Function declarations: extern int myStrl(const char *);
  - Static Data declarations: extern int globCount;
  - Constant definitions: const int MAXLINLEN = 80;
  - Comments that DOCUMENT the interface.

Header files MUST NOT contain:
  - Function DEFINITIONS: int myStrl( ... ) { ... return n; }
  - Simple Static Data DEFINITIONS: int globCount;
  - Aggregate DEFINITIONS: char hello[]="Hello";


Slide 4

Body files MUST CONTAIN:

- `#include` for the INTERFACE this body implements.
- `#includes` for the INTERFACEs of EVERY module this body USES.
- The DEFINITIONS of functions implemented by this module.
- DEFINITIONS of any static, global variables belonging to this module: 
  ```c
  int globCount = 0; char hello[]="hello";
  ```

Body files MUST NOT CONTAIN:

- Data type DEFINITIONS for interfaces (that should be in the header file).
- Data or function DEFINITIONS that do not belong to the ONE module this body implements.

Slide 5

Body files MAY CONTAIN:

- Constant or other definitions/declarations private to this module’s implementation.