

CSci 243

Discrete Structures

Project 2



Outline

- Project 2 Specification Tweaks to Ease Grading
- Modules and Interfaces
- README



New Requirements For Project 2

- You must use the project 2 template
<http://www.cs.wm.edu/~coppit/csci243/project2/project2-template-0.10.tar.gz>
- You must take filename on the command line
- Your output must be in the right format
- You must pass "make test"
- Programs must be named "project1" and "project2"



Modules In Various Disciplines

- Automotive Engineering
 - Engine, transmission, wheels, panels, chassis
- Civil Engineering
 - I-Beams, rivets, bricks, basement walls
- Electrical Engineering
 - Diodes, resistors, embedded chips, FPGAs
- Software Engineering
 - STL, language constructs, ???



Interfaces and Implementations

- Interfaces
 - Collection of public data and operations for a module
 - A contract between the module maker and the client
 - A precise definition of module's capabilities and use
- Implementations
 - Contains the ugly details of how to do the work
 - Hide performance optimizations, helper funcs, etc.
 - Can be changed without affecting clients



Modules

- A subsystem of the overall system
- Interface(s) and implementation
- Explicit dependencies on other modules (should)
- Hierarchical: modules within modules
- Precondition for components



Benefits of Modularity

- Parallel development
- Easier testing
- Easier innovation
- Conceptual clarity, concept limitation
- Faster compilation
- Reuse
- Access control



Module Examples In Software

- C: Function library with .h file and .c file
 - .h function declarations, .c function definitions
 - Missing: namespaces, structure, independ. meaning
- C++: Classes, separate interface & implement.
- Perl: Like C++, but no access enforcement or separate interface file
- COM: Binary objects, multiple rigid interfaces
 - QC: Why rigid?



Example: STL set

- Defines types
 - `set<int>::iterator`
- Defines methods on objects
 - `myset.insert(5)`
- Defines functions
 - `bool operator==(set<int> set1, set<int> set2)`
- Interface vs. Implementation
 - `.h` file, but no `.cc` file because of C++ limitations



README

- What is it in broad terms?
- How do you configure it for building?
- How do you build?
- How do you install it?
- How do you configure it for running?
- Where is further documentation?
- What's in the files/directories?
- Where to download new versions?
- Who wrote it?
- Are there any known bugs/issues?
- What libraries, OSes, etc. does it depend on?
- What software license?
- What remains to be done?
- How is it designed?



Stay Tuned

- To be updated/uploaded by tomorrow:
 - New project requirements
 - Makefile

