SE 2A04 Fall 1999

Modules

Instructor: W. M. Farmer

Revised: October 19, 1999

Module Components

1. Interface

   - **Specification** of the programs which are accessible outside of the module

2. Internal structure

   - **Implementation** of the module's interface
   - May be organized into submodules

Interface Components

1. **Language** of **interface programs**

   - Interface programs can be procedures, functions, objects, services, etc.

   - Interface programs are intended to work together

   - Interface programs are "exported" to other modules

   - A program of the module is part of the interface iff (if and only if) it is accessible outside of the module

2. **Assumptions** about the behavior of the interface programs

   - Also called **axioms**

Internal Structure Components

1. **Language** of **internal programs** and interface programs "imported" from other modules

   - Internal programs are implemented in the module
   - Imported interface programs are implemented in other modules

2. **Definition** of the exported interface programs

   - Defined in terms of the internal programs and imported interface programs
   - An exported interface program may also be an internal program

Example: Lab 1 and Lab 2

- Program is composed of two modules

- Overlap module
  - One interface program: `overlap`
  - Internal structure may contain internal programs for checking whether A and B are both circles, are disjoint, are tangent, etc.
  - Module does not use any imported interface programs

- User interface module
  - One interface program
  - Internal structure may contain internal programs for requesting and validating input
  - Module uses one imported interface program: `overlap`

Example: Linked Lists

- Two modules
  - Share an interface
  - Have different internal structures

- Interface contains a language of nine interface programs
  - Many other programs can be built from the interface programs

Module Correctness

- A module is correct if its implementation (internal structure) meets its specification (interface)

- More precisely, a module is correct if the definitions of the interface functions satisfy the interface assumptions

Modifying the Internal Structure

- Must verify that the new definitions of the interface programs satisfy the interface assumptions

- Other modules do not need be modified

Removing an Interface Program

- Internal structure does not need to be modified

- Other modules using this interface program may need to be modified

Adding an Interface Program

- Ideally, the new program can be defined in terms of the module's internal programs and imported interface programs

- Care should be taken if the internal structure must be modified

- Other modules do not need to be modified

Modifying an Interface Program

- Ideally, the new program can be redefined in terms of the module's internal programs and imported interface programs

- Care should be taken if the internal structure must be modified

- Other modules may need to be modified