

Procedural Abstraction

EE 564

Lecture 3

Daqing Hou, Winter 2007

Today:

- Procedural abstraction
- Benefits of specifications
- Procedure design

Procedural abstraction

- Familiar to everyone
- What is a procedure in Java?
- What can a procedure do?

Benefits of specifications

- Locality
 - Implementation of one abstraction can be read or written without examining the impl. of other abstractions
- Modifiability
 - Abstraction can be re-implemented without requiring changes to any abstractions that use it
- Fine prints: provided that we have identified the right abstractions.

Procedure design

- **Minimality**
 - one spec is more minimal than another if it contains fewer constraints on allowable behavior; algorithms
- **Underdetermined behavior**
 - for certain inputs spec allows more than one possible impl.
- **Deterministic implementation**
 - nondeterministic behavior is enabled by random number generator or system clocks
- **Generality**
 - more general if a spec can handle a larger class of inputs
 - e.g., parameterization on array size

Procedure design

- Total procedures
 - behavior defined for all inputs; empty requires clause
 - safe to use
- Partial procedures
 - unsafe; requires clause non-empty
 - use in local context or
 - where they enable substantial benefit, such as better performance
- Impl. may check constraints in requires clause when checking is not expensive.