Notes 1, Text: Ch. 1 (cont)

- Historical trends: from expensive hardware and few users to cheap hardware and many users.
- The Software Problem: We need formal solutions to informally described problems.
- SW development myths: adding people to late projects, changing software is easy, ...

Notes 1, Text: Ch. 1, and Ch. 2, Assignment 1

- Software processes:
  - Waterfall model: phases and limitations.
  - Evolutionary model.
  - Incremental development.
  - Spiral model.
  - Unified development process.
  - Agile development, for example
    - Extreme programming.
    - SCRUM
  - Test First Development (A1)
- Software disasters and successes.

Notes 2, Text Ch. 3-8 (pp. 67-238) Assignment 2

OO Design and Implementation Concepts.
- Review object-oriented concepts.
  - Objects:
    - Hidden: state representation and method implementation.
    - Public: behavior access through public interfaces.
  - UML class diagrams.

Notes 2, Text Ch. 2, 5, 11 Assignment 2

- Associations:
  - Non-hierarchical.
  - Composition/aggregation - whole-part.
  - Specialization/generalization (inheritance), polymorphism.
  - Use dependencies.
- Relationship between designs and code:
  - Creating objects and their associations.
  - Designing and implementing.
Verification & Validation

- Notes 3, Text Ch. 8 (pp. 240-269), A1 & A3.
- Black-box class or cluster testing.
  - Test based on ordering of class objects.
  - Test based on operations.
  - Test multiplicity.
  - Use a test oracle.
- Test plans: test names, strategy, description, verification.
- Test driver design: oracles, catching exceptions, reporting results.
- Use of fault models in testing.

Fault & Failure RIP Model:

Conditions necessary for a failure to be observed
1. Reachability: Program location that contain the fault must be reached.
2. Infection: The state of the program must be incorrect.
3. Propagation: The infected state must propagate to cause some output of the program to be incorrect.

Tests and test paths:
Many tests can "cover" one test path.

Inspections:
- Performed on all kinds of software documents.
- Focus on goals --- finding errors.
- Participant roles.
- Procedures.

The need for simplicity in all software documents: code, designs, specifications, ...