Life Cycle of Software

Phase 1: Specification

- Clearly specify all aspects of the problem
- Optional: Produce a **Prototype Program** that simulates a solution

Phase 2: Design

- Divide into **Modules**
- Loosely Coupled, Cohesive modularity
- Develop **Interfaces**
- Parameters, PreConditions and PostConditions

Phase 3: Risk Analysis

- Identify, Assess and Manage "risks" associated with the project

Phase 4: Verification

- Goal: **Prove** that your solution or algorithm is correct
- **Assertions**: Statement about a particular condition in an algorithm
- Identify **Invariants**: Condition that is always true
- Usually: "Prove" informally

Phase 5: Coding

- Bottom-Up vs. Top-Down

Phase 6: Testing

- Verify assertions and invariants
- Validate modules

Phase 7: Refining the Solution

- Efficiency
- Clarity
- Generality

Phase 8: Production

- Distribute to intended users
- Archive

Phase 9: Maintenance

- Fix errors
- Enhance features

For most "larger" software projects,
➔ Phases 1-8 comprise 30% of the overall work
➔ Phase 9 (Maintenance) comprises 70%.