Concepts

• Modeling and models.
• Modeling languages such as UML
• System, software, and model
• Software engineering development activities
  – requirements elicitation
  – analysis
  – system design
  – object design
  – implementation
  – testing
Concepts (cont.)

Project Management
Handle project schedule
  Gantt chart
  Pert chart
UML

• Use case diagram
  – Include
  – Extends
  – generalization
• Sequence diagram
• State chart diagram
• Class diagram
  – Associations
  – Generalization/specialization
  – Aggregation
  – Composition
• Deployment diagram
• Difference and interrelationship between different diagrams. For example: state chart diagram is for single object, sequence diagram is for interactions among multiple objects. How can these two diagrams be related to each other?
Requirement Analysis and Analysis Object

- Functions – define main functionalities based upon the problem description
- Non-functional requirements
- Based on a description of a system, analyze the main functionalities of the system, and then carry out object analysis
- Analysis Object
  - Control objects
  - Entity objects
  - Boundary objects
System Decomposition

- Subsystems and classes
- Coupling and cohesion
- Layers and Partitions
- Architecture styles
  - Repository
  - MVC
  - Client/Server
  - Peer to Peer
  - Three-tier
Address Design Goals

• Hardware/Software Mapping – deployment diagram
• Persistent Data Management
  • Database
  • Files
  • Data structure
• Global Resource Handling and Access Control
  • Access matrix
• Global Control Flow
  • procedure-driven control
  • event-driven control
  • threads
• Boundary Conditions
  • Start up and shutdown
  • Exception handling
  • configuration
Object Design

- Delegation and inheritance
- Interface inheritance and implementation inheritance
- Design patterns
  - Adapter
  - Composite
  - Proxy
  - Facade
Class Interface Specification

• Type
• Signature
• Visibility
Testing

- Test cases
- Test stubs and drivers
- Black-box testing – test against specification
  - Equivalence testing
  - Boundary testing
- White-box testing – test against implementation
  - Path testing – use flow graph to calculate how many independent paths are in the program
    - $CC = \text{number of edges} - \text{number of nodes} + 2$
  - Loop testing
Testing (cont.)

- **Component Testing**
  - Unit testing
  - Integration testing
- **System testing**
  - Function testing
  - Structure Testing
  - Performance testing
  - Acceptance testing
  - Installation testing
Software Life Cycle

- Waterfall model
- V model
- Spiral model
- Issue-based life cycle model