Software Engineering I CS4310

**Test 1 Review**

**Fall 2018**

* Introduction (Chapter 1)
  + Definition of Software Engineering
  + Challenges of developing large systems
  + Reasons for project failure
  + Factors for project success
* Teams and Project Management (Handouts)
  + Personalities in teams
  + Elements of cooperative teams
  + Productive Meetings
* Requirements Engineering (Chapter 1 and 5)
  + Defining (Requirements definition and feasibility study)
  + Discovery (elicitation)
  + Analyzing (traceability—coverage, impact, derivation)
  + Communicating (requirements specification)
  + Qualifying (validation and verification)
  + Managing
* Requirements Analysis and Elicitation
  + Challenges
  + Approaches
    - Observation
    - Interview: models to facilitate interview
      * Use case model
      * Context models: identify people or organizations that influence others to get work done
      * Flow work model: flow of action between people and roles performed
      * Sequence work model: write steps or sequence of activities, triggers, and intent
      * Physical Site: Diagram the physical layout, location of things, hardware, and types of software
      * Artifact work model: structure and usage of a work artifact—key distinctions important to the work, strategies for construction and use, source of information in the artifact, and destination of artifact.
    - Apprenticeship
    - Ethnomethodology
    - Requirements Workshops
    - Scenario Exploration
  + Acceptance Criteria
  + Documentation
  + Use Scenarios
  + Stakeholder workshop
  + Stakeholder interviews
  + Identification of stakeholders
* Interview Report
* Requirements Verification and Validation
  + Errors, faults (defect), and failures
  + Sources (Interview report, scenarios, prototype, simulations & models, and requirements specification)
  + Techniques (Walkthroughs, review, checklists, inspections, and audits)
* Modeling
  + Use-case Modeling
    - Purpose
    - Elements: Actor, Use case (events), Relationships (include, extend, generalization)
    - Scenarios
  + Object-oriented model: Class Diagram (represents classes, their attributes and operations, and relationships between classes)
    - Class (name, attributes, and operations)
    - Characteristics of association (name, multiplicity-number of objects connected across an instance of association, and role-role class plays in the relationship)
    - Interrelationships (Inheritance, Association, Aggregation, and Composition)
* Principles of Software Engineering (abstraction, anticipation of change, generality, modularity, separation of concerns)
* Feasibility Study and Report