IC470, Software Engineering

Exam: In class as per the syllabus

- As per the course policy, only non-programmable calculators may be used on the exam
- The exam will primarily cover chapters 7, 14, and 15 of the Schach text, but necessarily builds on the material covered in the milestones as well as in the first six weeks of the course
- The following exam objectives are intended to aid you in your preparations for the course's examinations However, the following are not considered to be inclusive of all the testable material from this course All assigned reading material, supplemental handouts, class lectures, class discussions, homework, labs, programming projects, team assignments, and in-class problem-solving sessions are considered testable material

========== 12 Week Exam Objectives ==========

Project Planning

- Be able to read a Gantt chart
- Understand the project planning information communicated by a Gantt chart
- Describe the use of a Gantt chart as a project planning tool, both for planning and for monitoring actual progress during project development
- Be able to construct a Gantt chart given estimates of the time required to complete the various tasks of a project and the software development method being used
- Understand the impact on project planning and Gantt chart construction of various software lifecycles such as Waterfall, Spiral Model, and Agile

Object Oriented Design (OOD)

- Understand the role of scenarios within OOD
- Describe the role of messages exchanged between class instances on UML sequence diagrams
- Be able to use the UML sequence diagram notation as part of the OOD of a software requirement
- Be able to evaluate the correctness of a UML sequence diagram
- Be able to use the UML detailed class diagram notation as part of the OOD of a software requirement
- Be able to evaluate the correctness of a UML detailed class diagram
- Apply the steps of OOD as presented by the author to include UML sequence diagrams, detailed class diagrams to include methods as class members that appeared on the sequence diagrams, detailed design - pseudo-code
- Understand metrics available during OOD

Cohesion and Coupling

- Define a module in terms of both structured and object-oriented programming

- Describe cohesion
- Understand the levels of cohesion
- Characterize the cohesiveness of a module
- Describe coupling
- Understand the levels of coupling
- Characterize the coupling between modules
- Understand the impact of object-oriented inheritance on coupling and cohesion
- Understand software reuse regarding modules

Dynamic Testing

- Understand why the number of all possible test cases rapidly becomes unmanageable/impossible when testing to specifications (black box) and testing to code (glass box)
- Apply equivalence class techniques for reducing the number of test cases
- Use boundary value analysis (edge cases) in test data set construction
- Be able to develop black box test cases that focus on testing the specification of a module
- Be able to develop glass box test cases that focus on statement, branch, or path coverage of given source code
- Be able to determine the number of test cases needed for complete path, branch or statement coverage of given source code.
- Apply McCabe's cyclomatic complexity metric to determine the complexity value (M) of given source code
- Understand the relationship between McCabe's Metric and the test cases needed to ensure branch coverage of a module
- Determine whether a module should be redesigned based upon its McCabe's Metric (M) value

ACM/IEEE Software Engineering Code of Ethics ("Software Engineering Code of Ethics is Approved" *Communications of the ACM*, October 1999/Vol 42, No. 10, pp 102-107.

- Understand the principles of the Software Engineering Code of Ethics
- Apply one or more principles of the Software Engineering Code of Ethics to a given scenario in which one or more principles are potentially violated. Be able to identify the specific principle(s) and explain why the principles were violated
- Comprehend why the Software Engineering Code of Ethics was developed