CSE 3411 Software Testing 1 (3 credits)

Primary Instructor: Cem Kaner **Supporting faculty:** Keith Gallagher

Textbooks and references: (None)

Course information:

2014–2015 Catalog description: CSE 3411 Software Testing 1 (3 credits). Explores functional (black box) methods for testing software systems, reporting problems effectively and planning testing projects. Students apply what they have learned throughout the course to a sample application that is commercially available or under development. The choice of sample application changes from term to term. Prerequisites: CSE 1002, CSE 1400 or ECE 2552, ECE 3541.

Prerequisites by topic: Object-orieuted program design concepts; use of exception handling to detect and resolve run-time errors; performance analysis; use of built-in Java methods for parsing; formatting and converting data; understand numbering systems, counting concepts, logic, sets, relations, and functions

Place in program:

Computer Science Program: Advanced elective

Software Engineering Program: Required

Course outcomes & related student outcomes: The student will be able to

1. Build a basic foundation in test techniques: knowing what they are and how to use them. (1: Fundamental knowledge & 2: Scientific, computing, and engineering problem solving)
2. Understand the basic skills for project-by-project analysis and trade-offs to determine the best testing strategy for a project. (2: Scientific, computing, and engineering problem solving & 4b: Satisfaction of requirements)
3. Foster strategic thinking, such as prioritizing and weighing costs or risks against benefits. (7: Communicate effectively)
4. Apply and extend critical thinking skills, such as analyzing, troubleshooting, and developing conclusions that can be justified. (7: Communicate effectively)
5. Apply and extend communication skills, especially clear and persuasive technical writing. (7: Communicate effectively)
6. Practice techniques for active reading, ambiguity analysis, and other skills for reading documents critically and efficiently and for taking tests well. These are important for handling increasingly difficult classroom assignments aud also for competent testing (such as deriving test cases from specifications). (9: Continually learn)
7. Improve and apply teamwork skills (peer reviews, paired testing, and shared analysis of challenging problems). (8: Effective teamwork)
8. Improve abilities as a technical investigator. (2: Scientific, computing, and engineering problem solving)

9. Apply/extend ability to think about a product or project from multiple viewpoints, such as thinking of different types of risk scenarios or using scenarios for a product. (4c: Trade-offs in design choices)
10. Improve study and reading skills. (9: Continually learn)
11. Create materials to be used for job interviews; plan for job interviews as part of the learning activity in the course. (9: Continually learn)

Topics covered:

1. Software testing techniques (9 hours)
2. Bug advocacy (9 hours)
3. Requirements analysis for test documentation and test antomation (9 hours)
4. GUI test automation tradeoffs (6 hours)
5. Issues in test automation architecture (3 hours)
6. Exploratory versus scripted testing (3 hours)

Approved By: Cem Kaner, Professor & Keith Gallagher, Associate Professor, Director of Software Engineering Programs

Signature: Cem Kaner **Date:** 2/2/15

Signature: Keith Gallagher **Date:** 10 F 15