CSCI 3350 Final Exam – Topic Outline

Fall 2014 Date: 8 December

Section 001: 8:00 – 10:00 am

Section 201: 6:00 – 8:00 pm

# Exam Format

Part 1 – Factoids: True or False, Fill-in-the-blanks, Multiple Choice

Read carefully, these questions are not designed to be deliberately misleading, but words have meaning. *(~25 points)*

Part 2 – Short answer questions: Brief, to the point answer, 1-3 **well-chosen** sentences, a simple diagram or sketch. *(~ 25 points)*

Part 3 – Calculations, Analysis, and Discussion (*~ 30 points)*

Part 4 – Design Patterns and CMMI (multiple choice) (*~13 points)*

Part 5 – Questions from the class presentations (multiple choice) (*~10 points*)

# Topics

The exam will cover material from the lectures, textbook, handouts and project. Questions in Part 1, 2, 4, and 5 will be taken from the material since the Midterm. **Questions in Part 3 are course comprehensive.**  **Any topic, which has been covered, is fair game.** The following is a high level overview of the topics covered since the midterm; **it is not intended to be an all-inclusive, detailed list of the test questions.**

1. Software Architecture *(Lecture 8)*
   1. Four basic software functions
   2. Three primary hardware components
   3. Architecture style is determined by the way in which the software functions are delivered over the hardware components
   4. Two categories of architectural styles
      1. Independent Components Style
         1. Server based
         2. Client based
         3. Client-server
         4. Multi-tiered
         5. Distributed Objects
      2. Data Flow Styles
         1. Pipes and Filters
         2. Batch Execution
2. Design Patterns *(Lecture 9)*
   1. Origin of Design Patterns
   2. Categories of Design Patterns
   3. Design Pattern Scope
   4. Details of the Eight Basic Design Patterns
      1. Adapter
      2. Bridge
      3. Composite
      4. Façade
      5. Iterator
      6. Observer
      7. Proxy
      8. Singleton
3. Implementation *(Lecture 10)*
   1. Choice of a programming language
   2. Characteristics of the generations of programming languages
      1. First Generation
      2. Second Generation
      3. Third Generation
      4. Fourth Generation
   3. General Practices
   4. Coding Standards
   5. Definitions of error, failure, fault
   6. Lines of Code as an Implementation Metric
   7. Buffer Overflow Attacks
      1. Prevalence of buffer overflow attacks
      2. Describe how a buffer (or string) overflow is the mechanism for stack smashing
      3. Describe how a smashed stack can lead to execution of attacker supplied code
      4. Techniques for minimizing the possibility of stack smashing attacks
   8. Origins of Bad Software
      1. Technical
      2. Psychological
      3. Real world
   9. Viega and McGraw’s *Ten* *Guiding Principles for Software Security*
      1. Example of each and how to mitigate
4. Capability Maturity Model *(Lecture 11)*
   1. What is the SEI and what is their role?
   2. What is the Capability Maturity Model Integrated
   3. Why all the emphasis on process?
   4. The Maturity Levels
5. Maintenance *(Lecture 12)*
   1. Issues
      1. What is maintenance
      2. Why does the software engineer care about maintenance?
      3. What is the challenge for the maintenance team?
   2. Four categories of maintenance
   3. Role of testing during maintenance
   4. Development process during maintenance

# Review carefully

1. Test 1 for Calculations and Discussions
2. In class exercises and homework problems
3. In-class presentations of Joel essays
4. Class notes

**Examples**

An essential component in the demonstration of competence in a subject is the ability to provide concrete examples. Be prepared to give specific examples of all general principles, guidelines, and methods that have been discussed during the semester.