

Final Exam Review

CS 3398
Spring 2013

Jill Seaman

1

Final Exam

- Section 251(MW):Wed, May 8, 2:00PM to 4:30PM
- Section 252(TR):Tues, May 7, 11:00AM to 1:30PM

- Closed book, closed notes, clean desk
- Chapters 4 through 9
- 25% of your final grade
- I recommend using a pencil (and eraser)
- I will bring extra paper.

2

Exam Format

- Multiple choice: 17 questions (ch 6-9)
- Problems: 3
 - write or modify some requirements (ch 4, SRS)
 - draw some diagrams/models: in the context of
 - system architecture (ch 6) and/or
 - design+implementation (ch 7, but review ch 5)
- Written answers: 3
 - ch 5-9, maybe SRS related
 - 3 to 5 sentences, generally
 - Define, explain, compare, evaluate
 - Make claims and give support

Each question will indicate how many points it is worth (out of 100)

3

Ch 4: Requirements engineering

- Requirements
 - Business, user, system
 - Functional vs non-functional
 - Qualities: complete, correct, clear, unambiguous, verifiable
 - Know how to write them
- Software Requirements Specification
 - Sections (generally, at least the 3 main sections)
 - Uses
 - Be familiar with yours

4

Ch 5: System modeling

- UML Models:
 - activity diagram
 - **use case diagram**
 - sequence diagram
 - **class diagram** (Aggregation and generalization)
 - state diagram
- How models are used
 - Requirements development, design and implementation
- Be able to
 - Recognize the models
 - Draw simple versions of the models

5

Ch 6: Application architecture

- Introduction
 - Terms: Architectural design, Software architecture
 - Using box and line diagrams
- Design decisions
 - standalone? distributed? coupling and cohesion
 - architecture affects non-functional requirements
- Architectural patterns
 - ModelViewController - Client-Server - Repository
 - Layered - Pipe & Filter
- Application architectures
 - Transaction processing systems
 - Language processing systems

6

Ch 7: Design and implementation

- Object oriented design activities
 - Requirements elicitation (use cases)
 - Object Oriented Analysis (object model)
 - System Design (architecture)
 - Object Design (class diagram)
 - Implementation (map diagram to code)
- Design patterns
 - What are they, how described, why used?
 - Observer pattern: be familiar with this one
- Remaining issues:
 - Reuse: benefits+costs,
 - Configuration management: why version control?
 - Open source development: pros/cons, licensing issues

Also: Interface Specifications

7

Ch 8: Software Testing

- Verification and Validation
 - static vs dynamic verification
 - concepts, Test cases, white box vs black box testing
- Testing activities
 - Development
 - * Unit * Component * System
 - Release
 - User
 - * Alpha * Beta * Acceptance
- Techniques for choosing test cases
 - Partition - Path Testing
 - Guideline-based - State-based testing
 - Use case-based, Scenario, and Requirements-based testing

8

Ch 9: Software evolution

- Evolution Process
 - Spiral model: iterative development
 - Steps to next Release, Change Implementation
 - Program understanding
- 3 Types of software maintenance
 - Defect fixing, adapting to new environment, new features
- Reengineering
 - What, when, why (why not start from scratch?)
 - Techniques
- Refactoring
 - What, when, why
 - Bad smells

9

Example Problems

- See TRACS for the in-class modeling exercises (and solutions) from chapter 6.
- Note: read multiple choice questions carefully.
 - Treat a question like the following as a series of T/F questions

Which of the following is NOT a characteristic of the Scrum method?

- T (a) There is no real project manager, the team makes its own decisions.
- T (b) The stakeholders select the features for each sprint cycle.
- T (c) There are daily meetings where each developer gives a short report about their work.
- F (d) The scrum master assigns the tasks to the developers at the beginning of the sprint cycle.

10

Office Hours

Day	Date	Time
W	5/1	12:00-1:30pm
Th	5/2	3:00-4:30pm
M	5/6	12:00-1:30pm
W	5/8	12:00-1:30pm
		and by appt.

11