Final Exam Review

CS 3398
Spring 2012
Jill Seaman

Final Exam

- Wednesday, May 9 (2-4:30pm) for .251 and Tuesday, May 8 (11-1:30) for .252
- 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.

Exam Format

- Multiple choice questions
- Problems
  - write (or modify) some requirements
  - draw some diagrams/models: in context of system architecture and design+implementation
- Written answers
  - 3 to 5 sentences, generally
  - Define, explain, compare, evaluate
  - Support with three reasons, unless stated otherwise
- Each question will indicate how many points it is worth (out of 100)

Example Problems

- I will post some on the class website by Tuesday, May 1.
Ch 4: Requirements engineering

- Requirements (define)
  - Business, user, system
  - Functional vs non-functional
  - Qualities: complete, correct, clear, unambiguous, verifiable

- Requirements Development (and management)
  - Elicitation, Analysis, Specification, Validation (interleaved)
  - Goal: Software Requirements Specification, uses of it
  - Stakeholders and analysts

- Tools, methods
  - interviews, elicitation workshop, ethnography
  - Scenarios, use case diagrams, modeling
  - Prototypes, requirements review, generate test cases

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

Ch 6: Application architecture

- Introduction
  - Terms: Architectural design, Software architecture
  - Using box and line diagrams

- Design decisions
  - 3 questions to ask
  - how architecture affects non-functional requirements

- Architectural patterns
  - ModelViewController - Client-Server - Repository
  - Layered - Pipe & Filter

- Application architectures
  - Transaction processing systems
  - Language processing systems

Ch 7: Design and implementation

- Object oriented design activities
  - Define system context and interactions (external!!)
  - Design system architecture
  - Identify principal objects
  - Develop design models: class, sequence, state (as needed)
  - Specify interfaces

- 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
Ch 8: Software Testing

- **Verification and Validation**
  - software inspections and software testing

- **Stages and types**
  - Development
    - Unit
    - Component
    - System
  - Release
  - User
    - Alpha
    - Beta
    - Acceptance

- **Techniques for choosing test cases**
  - Partition
  - Guideline-based

- **How to test a class**
  - attributes
  - operations
  - states

Ch 9: Software evolution

- **Evolution Process**
  - Spiral model: iterative development
  - Driven by change requests
  - Program understanding

- **3 Types of software maintenance**
  - Defect fixing, adapting to new environment, new features

- **Reengineering**
  - What, when, why + techniques

- **Refactoring**
  - What, when, why + bad smells

- **Legacy system management**
  - 4 strategies: scrap, maintain, reengineer, replace
  - Assessment: business value/system quality

Office Hours

<table>
<thead>
<tr>
<th>Day</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>4/30</td>
<td>2-3pm, 3:30-4:45pm</td>
</tr>
<tr>
<td>T</td>
<td>5/1</td>
<td>None</td>
</tr>
<tr>
<td>W</td>
<td>5/2</td>
<td>2-3pm</td>
</tr>
<tr>
<td>Th</td>
<td>5/3</td>
<td>11am-noon</td>
</tr>
<tr>
<td>M</td>
<td>5/7</td>
<td>2-3pm</td>
</tr>
<tr>
<td>T</td>
<td>5/8</td>
<td>None (exams)</td>
</tr>
<tr>
<td>W</td>
<td>5/9</td>
<td>None (exam)</td>
</tr>
</tbody>
</table>

And by appointment