EECE 493 Software Engineering (Final Exam Study Guide)
7/26/2012
FINAL EXAM : Wednesday, August 1 (12:30 – 2:00), 661 Baldwin

The final exam will be comprehensive, covering all the material discussed in class for the course. This study guide is intended to help focus your study time. There is no guarantee that it is complete. You are responsible for all the material covered in class and included in the notes posted on blackboard.

In regard to the lecture material from textbook, listed below are the sections you should review. There is a case study at the end of each chapter. It will help in reinforcing the concepts covered in chapter but make sure you are clear about the concepts before picking them up.

1. Chapter 1 : Sections 1.2.1, 2.3.5
   : Section 1.4

2. Chapter 2: All except Sections 2.2.4 and 2.4.7 (State Machine diagrams and Communication Diagrams not included)

3. Chapter 4: All except Sections 4.5.1 and 4.5.3

4. Chapter 5: All except Sections 5.4.9 amd 5.4.12, 5.5

5. Chapter 6: All except 6.3.4 and 6.4.3

6. Chapter 7: All except Sections 7.4.5, 7.4.6, 7.4.7, and 7.5
   Section 7.6 presents a case study containing good examples of selection of a specific architecture style based on a specific design goal.

7. Chapter 8: Refer the handout distributed in class for the design patterns
8. Chapter 9: Class notes/blackboard, Sections 9.3.3, 9.3.4, 9.3.5, 9.3.6
9. Chapter 11: Everything till 11.4.5 (Exclude, state based and polymorphism testing)

Make sure you know the meaning of these terms:

1. Application domain vs Solution domain
2. Software development phases and major activities in each
3. UML diagrammatic notations (Use cases, class diagrams, sequence diagrams, activity diagrams)
4. Constructs for each UML diagram type (Include/extend/inheritance for use-cases, multiplicity constraints/association-names in class diagrams, looping, conditional constructs in sequence diagrams, join/fork in activity diagrams)
5. Software development lifecycle models (Waterfall, prototype etc.). Pros and cons of each
6. Functional and non functional requirements : What each of non-functional requirements mean

7. Desirable characteristics of requirement specification (Verifiability, unambiguous etc.)
8. Design goals
9. Cohesion and coupling
10. Architectural styles
11. UML deployment diagrams
12. What defines a subsystem: services , interfaces
13. Design patterns
14. OCL specifications
15. Testing strategies : refer notes and sections marked in textbook

Go through the in-class exercises, quiz and midterm.