

CS 6362.001 — Software Architecture and Design

Fall 2002

Instructor: Lawrence Chung
Office: ES 3.204, Phone: 883-2178, e-mail: chung@utdallas.edu
Office Hours: M: 01:30–02:30pm, or by appointment
Lectures: Time: MW 11:00am–12:15pm, Room: ES2.305
TA TBD

Prerequisites CS 5V81 Software Engineering

Primary Reading: *Software Architecture: Perspectives on an Emerging Discipline*
Mary Shaw and David Garlan, Prentice hall

References: *Software Architecture in Practice*,
L. Bass, P. Clements & R. Kazman, Addison Wesley
Component-Based Software Engineering,
Edited by A. W. Brown, IEEE Computer Society
Design Patterns: Elements of Reusable Object-Oriented Software,
Eric Gamma, Richard Helm, Ralph Johnson and John Vlissides,
Addison-Wesley
Design Patterns for Object-Oriented Software Development
Wolfgang Pree, Addison-Wesley Longman
Seamless Object-Oriented Software Architecture:
Analysis and Design of Reliable Systems,
Kim Walden & Jean-Marc Nerson, Prentice Hall
Designing Enterprise Applications with the J2EE Platform, 2/E
Inderjeet Singh, Beth Stearns, Mark Johnson, The Enterprise Team, Addison We
Understanding CORBA: The Common Object Request Broker Architecture
Randy Otte, Paul Patrick and Mark Roy, Prentice Hall
The Essential Client/Server Architecture: Survivor's Guide
Robert Orfali, Dan Harkey and Jeri Edwards, John Wiley & Sons
Network Application Support for Building Open Systems
James Martin and Joe Leben, Digital Press
Non-Functional Requirements in Software Engineering
Lawrence Chung, Brian Nixon, Eric Yu and John Mylopoulos,
Kluwer Academic Publishing

Grading Scheme: Projects (Sep. 30(M), Nov 11(M)): $2 \times 20\% = 40\%$
Term Paper (Nov 18(W)): 15%
Test (Nov 25(M)): 40%
Class Participation: 5%

Course Project:

The project will be done by teams of 3 students. (Teams with more or less than 3 members will be allowed only under exceptional circumstances). All students in a team will get the same mark for the work they do unless they unanimously agree (in writing) to an unequal division. You are to choose your own team members. No late projects. An orphan will be assigned to a team by the instructor.

Course Outline:

Concepts and methodologies for the systematic analysis, development, evolution, and reuse of software architectural design. Common software architectural styles, elements and connectors. Decomposition and composition of software functionality. Non-functional requirements as criteria for analyzing trade-offs and selecting among architectural design alternatives. State of the practice and art.

1. Introduction to Software Architecture
2. Classical Module Interconnection Languages
3. Abstract Data Types
4. Modular Decomposition Issues
5. Data Flow
6. Repositories
7. Events
8. Process control
9. Client Server Architecture
10. Middleware
11. Design Patterns
12. Other Architectural Design Topics: DSSAs, System Integration, Processes, System Architecture, Process Architecture
13. Retrospective and Prospective

Course Policies:

- The University of Texas System Policy on Academic Honesty (*The Regents and Regulations, Part ONE, Chapter VI, Section 3, Paragraph 3.22*):

Any student who commits an act of scholastic dishonesty is subject to discipline. Scholastic dishonesty includes but is not limited to cheating, plagiarism, collusion, the submission for credit of any work or materials that are attributable in whole or in part to another person, taking an examination for another, any act designed to give unfair advantage to a student or the attempt to commit such acts.

The minimum penalty for academic dishonesty is a failing grade (zero) for the project or test.

- It is your responsibility to attend all the classes, to participate in class discussions, understand any announcements, and follow changes to this course syllabus.

Online Course Material:

All continuously evolving viewgraphs and project descriptions can be accessed through:

<http://www.utdallas.edu/~chung/SA/contents.html>