

CSE 4233
Software Architecture and Design

REQUIRED/ELECTIVE:

Computer Science – Elective
Software Engineering – Required
Computer Engineering - Elective

CATALOG DATA:

(Prerequisite: CSE 4214/6214 with a C or better). Three hours lecture. Topics include software architectures, methodologies, model representations, component-based design, patterns, frameworks, CASE-based designs, and case studies.

PREREQUISITE BY TOPIC:

1. Computer programming
2. Software design
3. Software development life cycle
4. Object-oriented analysis and design

TEXTBOOKS AND OTHER REQUIRED MATERIAL:

Paul Clements, Felix Bachmann, Len Bass, et al. *Documenting Software Architectures: Views and Beyond*, Boston: Addison-Wesley, 2003, ISBN 0-201-70372-6.
Handouts and website www.cse.msstate.edu/~cs4233

COORDINATOR:

Dr. Jeff Carver

COURSE OBJECTIVES:

1. Understanding of software architecture principles and commonly used styles. (At the architecture level, “patterns and frameworks” are called “styles”.)
2. Ability to analyze a software architecture as a set of views of a model.
3. Ability to design software architectures that are well-suited to requirements, constraints, and concerns of stakeholders.
4. Independent learning of the Unified Modeling Language (UML) and the ability to perform CASE-based design of software architectures using UML.
5. Ability to document work to an acceptable standard.
6. A basic understand of how to review software documents to detect problems.

TOPICS COVERED:

(Number of class hrs)

1. What is “software architecture”	3
2. Views of architectures	4
3. What is an “architecture description”	4
4. Architecture methodology	4
5. Unified Modeling Language (UML) overview	2
6. Inspection of software architectures	3
7. Module Viewtypes	4
8. Component-and-connector Viewtypes	6
9. Allocation Viewtypes	3
10. Process control styles	3
11. Advanced topics	5
12. Tests and review	4

CONTRIBUTION TO PROFESSIONAL COMPONENT:

Engineering Topics of Engineering Science and Design.

ASSESSMENTS:

1. Short quizzes in lecture meetings and tests
2. Creation of software architecture document in homework assignments

RELATIONSHIP TO PROGRAM OUTCOMES:

Note: Parenthesized list indicates the ABET criteria, Computer Engineering outcomes, and Software Engineering outcomes addressed by each performance criteria.

Performance Criteria:

1. Understanding of software architecture principles and commonly used styles. (a, se3)
2. Ability to analyze a software architecture as a set of views of a model. (a, se3)
3. Ability to design software architectures that are well-suited to requirements, constraints, and concerns of stakeholders. (c, e, h, se3, se5, se8)
4. Independent learning of the Unified Modeling Language (UML) and the ability to perform CASE-based design of software architectures using UML. (k, se7)
5. Ability to document work to an acceptable standard. (g, se2)
6. A basic understand of how to review software documents to detect problems. (se3)

PREPARED BY:

Jeff Carver, Assistant Professor, Department of Computer Science & Engineering, March 15, 2004;

ESTIMATE CSAB CATEGORY CONTENT:

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	_____	_____	Computer Organization and Architecture	_____	_____
Algorithms	_____	_____	Concepts of Programming Languages	_____	_____
Software Design	_____	_____ 3			

ORAL AND WRITTEN COMMUNICATIONS:

Every student is required to submit at least 3 written reports (not including exams, tests, quizzes, or commented programs) of typically 10-15 pages and to make 0 oral presentations of typically _____ minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

SOCIAL AND ETHICAL ISSUES:

The effects of business and social concerns on technical aspects of architecture design are discussed in the context of case studies

THEORETICAL CONTENT:

1. Software architectures styles
2. Analysis methods for software architectures

PROBLEM ANALYSIS:

One type of homework assignment requires the students to review an existing software architecture document to look for problems. The second type of homework assignment requires the students to analyze the requirements from a realistic hypothetical system to determine how to develop the software architecture for that system.

SOLUTION DESIGN:

During the homework assignments, the students are required to use the information gathered from their analysis of system requirements to develop a feasible software architecture for the system.