

CS 532 - Software Design - 3 Credit Hours**Textbook (required):**

Software Design, *Budgen, David, Addison-Wesley, 1994* .

Software Architecture: Perspectives on an Emerging Discipline, *Shaw, M., and Garlan, D., Prentice Hall, 1996*.

Alternate References:

The Essence of Structured Systems Analysis Techniques, *Griffiths, Gary, Prentice Hall Europe, 1998*.

Bringing Design to Software, *Winograd, Terry, ACM Press and Addison-Wesley, 1996*.

Software Architecture and Design, *Witt, B.I., Baker, T.F., Merritt, E.W., Van Nostrand Reinhold, 1994*.

Object-Oriented Modeling and Design, *Rumbaugh, et. al., Prentice Hall, 1991*.

UML Distilled: Applying the Standard Object Modeling Language, *Fowler, Martin and Scott, Kendall Scott, Addison-Wesley, 1997*.

Other reference material may be presented in class (for which the student is responsible).

Textbook Coverage:

Software Design – Chapters 1- 15.

Software Architecture: Perspectives on an Emerging Discipline – Chapters 1- 8

Course Objective and Description:

Starting as long as 35 to 40 years ago, computer scientists have looked for ways to improve the process of creating and maintaining software. The term software crisis was coined to define the problem of software cost and complexity rising faster than our ability to meet the demand for new software as well as the demand for maintaining existing software. Some estimate that 70% of the cost of software systems is in its maintenance.

One key to building good software is the consistent application of a design methodology. The more care taken in the design process, the better the product will be and the easier it will be to maintain. A design methodology also makes it easier to consistently produce good designs and makes it easier for team members to communicate between themselves.

One purpose of this course is introduce design methodologies. We'll survey "traditional" design methods, including the Jackson Structured Programming (JSP) and Structured Systems Analysis and Structured Design (SSA/SD). We'll also learn a relatively new method developed from the world of object-oriented programming, called Object Modeling Technique (OMT). Through the semester, you will work in teams to produce designs using one of these methods. At the end of the course you will know:

- √ Overview of the software development process
- √ Principles of software design (models/abstractions, representations and practices)
- √ Basic concepts of analysis and design
- √ Software analysis and design using a design methodology
 - Jackson Structured Programming/Design
 - Structured Systems Analysis and Structured Design
 - Object Modeling Technique
- √ Characteristics of object systems
- √ Validation (of the above) through your own design, implementation and demonstration

Topical Outline:

<i>Topic</i>	<i>Reference</i>	<i>Lectures</i>
1. Course introduction, syllabus, overview of some topics, course objectives, Software Architecture and Design – <i>Assignment: Define Conceptual Integrity</i>	Instructor materials	3
2. The Nature of the Design Process – <i>Assignment:</i>	Ch 1	1
3. The Software Design Process – <i>Assignment:</i>	Ch 2	.5
4. Design in the Software Development Process – <i>Assignment:</i>	Ch 3	.5
5. Design Qualities – <i>Assignment:</i>	Ch 4	1
6. Expressing Ideas about a Design <i>and</i> Some Design Representations – <i>Assignment:</i>	Ch 5 – 6	1
7. Some Design Representations – <i>Assignment</i>	Ch 6	1
8. Structured Systems Analysis and Structured Design – <i>Assignment:</i>	Ch 10	2
9. The Rationale for Method– <i>Assignment:</i>	Ch 7	1
10. Design Strategies <i>and</i> A Brief Interlude– <i>Assignment:</i>	Ch 8	1
11. Jackson Structured Programming – <i>Assignment:</i>	Ch 9	1
12. Structured Systems Analysis and Structured Design – <i>Assignment:</i>	Ch 10	1
13. Jackson Systems Development – <i>Assignment:</i>	Ch 1	1
14. Object-Oriented and Object-Based Design – <i>Assignment:</i>	Ch 12	1
15. Some Other Systematic Approaches to Design – <i>Assignment:</i>	Ch 13	1
16. A Formal Approach to Design – <i>Assignment:</i>	Ch 14	1
17. The Evolution of Software Design Practices – <i>Assignment:</i>	Ch 15	1
18. Software Architecture and Architectural Styles– <i>Assignment:</i>	Ch 1-2	2
19. Case Studies and Tools for Architectural Design – <i>Assignment: Each team will present a (10-12 minute) case study and describe the most appropriate tool for their case study system from chapter 8.</i>	Ch 3 & 8	1
20. Shared Information Systems (may be skipped) – <i>Assignment:</i>	Ch 4	1
21. Architectural Design Guidance – <i>Assignment:</i>	Ch 5	1
22. Formal Methods and Specifications – <i>Assignment:</i>	Ch 6	1
23. Linguistic Issues – <i>Assignment:</i>	Ch 7	1
Total (1 lectures in-class exam, 3 for student presentations PDR/CDR/Demo) = 30		26

Grading Policy:

- There will be (closed book/notes) one regular class period exam plus a comprehensive final exam (likely will be a take home).
- There will be at least 4 homework assignments.
- Final grades are computed from the following weights

Project	30%	Exam 1	30%	Final	30%
Homework		10%	<i>(Homework average is the normalized average of all homework grades)</i>		

All grades are based on a decade scale from 0-100 as follows:

90-100	=	A	A linear shift (known as curving) may be applied to the final grade averages as a one-time scale at instructors discretion. Plus and minus grades may be given out on the final course grade at the instructors discretion. Note, a grade of 'C' (or lower) is considered failing for graduate students.
80-89	=	B	
70-79	=	C	
60-69	=	D	
< 60	=	F	

Policy on Missed Exams

- Absolutely no makeup exams are given without prior authorization or written proof (or its equivalent) that the student was prevented from participating. Unexcused missed exams result in a grade of zero for the exam.
- Excused absences from exams include: (1) personal emergencies, (2) work-related (with confirmation letter) (Taking an exam early may be possible for unavoidable, planned absences.)

Cheating on Exams or Homework

- Absolutely no cheating on exams or projects and homework will be tolerated. Students are encouraged to discuss concepts for homework (individually and in class); however, each student is expected to develop his or her own solutions. For further details on academic honesty the student is referred to the University Catalog.

Class Attendance

- Class attendance is not required in the sense that no roll will be taken; however, EACH STUDENT IS 100% RESPONSIBLE FOR ALL MATERIAL AND ANNOUNCEMENTS COVERED IN CLASS.

Late Drop

- Dropping of a class after the deadline listed in the class schedule is governed by departmental and college policy. The student must show documented evidence supporting reasons for a request to drop a class after the deadline. Each request is considered on an individual basis for determining acceptance.

Office Hours:

- Typically I will be available after the class meets, or by appointment. Official office hours for this are posted at my office or check my home page at URL: <http://www.cs.uccs.edu/~sheldon>). Other modes of communication with me will also be announced -- e-mail (preferred), voice-mail, etc.

Standards and Grading Criteria for Computer Projects or Homework:

- Homework must be clearly presented and complete. Do not turn in scratch work (or multiple attempts).
- Homework that is not software will not be accepted after the due date.
- Homework is due at the BEGINNING OF CLASS (not accepted later) on the due date.
- Questions about solutions to problems are encouraged, but may have to be handled individually since class time is at a premium.
- Each problem will be numbered as it is in the text or on assignment sheet (no not attempt to renumber problems).
- Each problem will be stated fully on your solution sheet, including all drawings and diagrams. This includes any sub-parts labeled a), b), etc.
- Make it clear where your solution begins. A recommended format is to label all parts of the problem presentation clearly (e.g., Problem statement, Assumptions, Solution).
- Your solutions must be clear, effectively organized, and effectively presented. Show the major logical steps in achieving a solution. If an answer is obvious by inspection, then so state.
- Do not turn in pages with irrelevant scratch work or marked-through work.
- You should treat your homework assignments with the same pride as you would a proposal to a customer.
- The rules for grading are the following: (1) quarter-credit for stating the problem, (2) quarter-credit for an honest effort at a solution. The rest of the credit is subject to the following rule.
- While grading, I grade one problem at a time (homework and exams). I give the highest score (may not be the maximum) to the best answer and assign scores for the rest based on how well their solution compares with the best. Some problems are less subjective and naturally your answer is simply correct or not.