SE 370 Software Structures (3 units)

Course Description
Introduction to mathematical foundations for software engineering modeling. Engineering economics for software such as software lifecycle and evaluating cost-effective solutions. Modeling foundations such as modeling principles, invariants, and formal notations. Models of concurrent systems, verification of models, and limitations of these techniques. Analysis fundamentals such as analyzing forms, correctness, dependability, and formal analysis. *Prerequisite: CS 111*

Course Materials
- *Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures* by Hassan Gomaa
- *Supplemental materials on discrete structures, logic, and computability by instructors*

Learning Objectives
Students that successfully complete the course will be able to:
1. Identify, analyze, and apply skills and professional standards in mathematical and engineering fundamentals, and engineering economics for software.
    - A brief introduction to mathematical foundations for software engineering (e.g., notation and common concepts and basic results in trees).
    - Engineering economics for software (e.g., considerations through software life cycle, evaluating cost-effective solutions).
2. Identify, analyze, and apply skills and professional standards in software modeling and analysis in terms of modeling foundations, types of models, and analysis fundamentals.
    - Software modeling foundations (e.g., modeling principles, invariants, mathematical models, formal notations).
    - Types of software modeling (e.g., information modeling, behavior modeling, and architectural modeling).
    - Analysis fundamentals for software engineering (e.g., analyzing forms, correctness, dependability, formal analysis).

Course Structure
The course includes a brief introduction to discrete structures in addition to software structures and modeling for large software projects. The course cultivates clear thinking and creative problem solving, trains in the construction and understanding mathematical notation and common concepts and basic results in topics such as relations and graphs as needed for software engineering modeling. The course covers basic modeling principles, properties and expression of models, syntax, semantics and pragmatics, illustrates and employs information, behavioral and structure modeling, applies model analysis including analyzing for completeness, consistency and correctness, demonstrates the concepts of traceability and interaction analysis, provides examples of heuristic, formal, prototyping and agile methods.

The course serves as a foundation course covering basic concepts and tools useful to our students. The course coverage is a core part of a software education and not usually included intentionally in a computer science education.

What is the format of the course?
The course involves three hours of lecture/activity per week. The goal of every class session is for individual students to be able to apply concepts-related and other skills in software engineering as soon as possible.

How are students assessed?
Homework – Homework assigned throughout the term apply and reinforce material from the course lectures and discussion. Homework assignments also encompass case studies discussed in class to illustrate the importance of discrete and software structures in determining success and failure of software projects.

Quizzes – Short quizzes completed before or during class.

Exams – Exams will be used to test the students' knowledge and capabilities regarding the course material.

Final Exam – A comprehensive exam of the topics covered in the course.

Grade Distribution

Homework	Quizzes	Exams	Final Exam
30%	15%	25%	30%

SCALE:
A, 93 or more; A-, at least 90 but less than 93;
B+, at least 87 but less than 90; B, at least 83 but less than 87; B-, at least 80 but less than 83;
C+, at least 77 but less than 80; C, at least 70 but less than 77;
D, at least 60 but less than 70
F, 0 to less than 60

Let's suppose that your average homework score is 90%, your average quiz score is 86%, your average exam score is 80%, and your final exam score is 85%. Then your numerical course grade is:
90 x 30% + 86 x 15% + 80 x 25% + 85 x 30% = 85.4, which is translated as B.

Course Policies

Academic Honesty
Students will be expected to adhere to standards of academic honesty and integrity, as outlined in the Student Academic Honesty Policy. All written work and oral presentation assignments must be original work. All ideas/material that are borrowed from other sources must have appropriate references to the original sources. Any quoted material should give credit to the source and be punctuated with quotation marks.

Students are responsible for honest completion of their work including examinations. There will be no tolerance for infractions. The instructor reserves the right to discipline any student for academic dishonesty, in accordance with the general rules and regulations of the university. Disciplinary action may include the lowering of grades and/or the assignment of a failing grade for an exam, assignment, or the course as a whole. Incidents of Academic Dishonesty will be reported to the Dean of Students. Sanctions at the University level may include suspension or expulsion from the University. (http://www.csusm.edu/dos/conduct/index.html)

Campus Writing Requirement
The University Writing Requirement of 2,500 words in this course will be exceeded by solving the problems on your homework, quiz, and test assignments.

Inform Your Instructor of Any Accommodations Needed
Students with disabilities who require reasonable accommodations must be approved for services by providing appropriate and recent documentation to the Office of Disabled Student Services (DSS). This office is located in Craven Hall 4300, and can be contacted by phone at (760) 750-4905, or TTY (760) 750-4909. Students authorized by DSS to receive reasonable accommodations should meet with me to ensure accommodations are met.

Cougar Care Network (CCN)
Our campus has implemented a new early alert system to support and promote students' academic and personal success. I may refer you through the Cougar Care Network (CCN) to get connected to campus support and resources to assist you.

Civility Campaign
The Civility Campaign, an effort led by the Dean of Students Office, defines civility to reflect the community values of CSUSM. The university strives to be a community demonstrating respect for oneself and for others, treatment of others with dignity, and behaviors which promote a physically and psychologically safe, secure and supportive climate enabling all community members to engage as full and active participants where the free flow of ideas are encouraged and affirmed.

COURSE TOPICS for SE 370

- Mathematical notation for sets, relations, machine diagrams, and graphs (such as trees and directed graphs
- Software quality attributes such as maintainability, modifiability, testability, traceability, scalability, reusability, performance, security, and availability
- Software life cycles used for developing software, including the waterfall, prototyping, iterative, spiral, and unified process, including compare and contrasts.
- Engineering economics for software such as considerations through software life cycle and evaluating cost-effective solutions
- Overview of software modeling and design method
- Overview of key design concepts
- Modeling foundations such as modeling principles, invariants, and formal notations
- Case modeling (e.g., requirements modeling, case description, case relationships, include relationships, extend relationships, structuring guidelines
- Types of software modeling (e.g., information modeling, behavior modeling, and architectural modeling).
- Modeling foundations (e.g., modeling principles, invariants, mathematical models, formal notations).
- Analysis fundamentals for software engineering (e.g., analyzing forms, correctness, dependability, formal analysis).