

SOFTWARE ENGINEERING CAPSTONE

CS 4704

I. CATALOG DESCRIPTION

Senior project course integrating software engineering knowledge and skills acquired in previous courses. Team-based approach to problem formulation, requirements engineering, architecture, design, implementation, integration, documentation and delivery of a software system that solves a real-world problem. Pre: C or better in 3704 (3H, 3C).

Course Number: 4704

ADP Title: Software Engineering Capstone

II. LEARNING OBJECTIVES

Having successfully completed this course, students will be able to:

- Design and implement a solution to a significant open-ended problem in software engineering.
- Document and present (using written, oral and visual means) the design process and the results from a solution to a problem in software engineering.
- Evaluate or assess a solution to a problem in software engineering, and compare with alternative approaches.
- Discuss and evaluate their contribution to a team project.

III. JUSTIFICATION

This course provides a capstone design experience for students interested in software engineering. While previous courses in the Computer Science curriculum provide students with the knowledge and skills necessary to begin a career as a software engineer, the end-to-end design and implementation experience offered in this course, done as part of a team, and addressing an open-ended problem, comes much closer to the experience that software engineering professionals encounter in the work place. The emphasis on documentation and presentation of designs and results is also valuable career preparation, and forces students to take ownership of their designs and systems as they present them to others.

This course is offered at the 4000 level because it requires a basic understanding of software design and software engineering principles and practice, as provided by the prerequisite (CS 3704, Intermediate Software Design and Engineering), and courses that are prerequisite to 3704.

This revision updates the course topics to reflect current software engineering practice, and includes emphases appropriate to a capstone course. The original course proposal was approved before Computer Science had a capstone requirement for its majors.

IV. PREREQUISITES AND COREQUISITES

Background in software design and software engineering, as taught in CS 3704 (Intermediate Software Design and Engineering) and its prerequisites, is essential for understanding and applying the advanced software engineering issues and practices that are integral to this course.

V. TEXTS AND SPECIAL TEACHING AIDS

Required:

Sommerville, Ian. SOFTWARE ENGINEERING. Addison Wesley, 2011, xvi, 792.

Additional on-line materials will be used based on the particular problem and approach used in a given semester.

VI. SYLLABUS

	Percent of Course
1. Problem Formulation	6%
2. Requirements Engineering.....	6%
3. Architecture.....	8%
4. Design	35%
• Design specification based on the architecture identified	
• Design verification, validation and quality assurance	
• Design specification document	
5. Implementation	35%
• Software development for modern platforms	
• Implementation verification, validation, and quality assurance	
• Program documentation	
6. Integration and Delivery	10%
	<hr/> 100%

VII. OLD (CURRENT) SYLLABUS

	Percent of Course
1. Software Life Cycle	10
a. General software engineering issues	
b. Software development Life Cycle	
c. Motivation for software engineering	
2. Software Design Languages	10
a. PSL/PSA	
b. Superman	
3. Software Design Methodologies	
a. Yourdan's composite design	10
b. Jackson's Methodology	7
c. Top down versus bottom up design	6
4. Software Testing Methodologies	
a. White box testing	10
b. Black box testing	10
c. Life cycle testing	10
d. Walkthroughs	7
5. Management of the Software Project	10
a. Scheduling tasks	
b. Timing of tasks	

6. Software quality metrics	10
a. Code metrics	
b. Structure metrics	
	<hr/>
	100

VII. CORE CURRICULUM GUIDELINES

NA