

Syllabus for CSC-382 Winter 2008

Instructor Information:

- Dr. Michael Farmer
- Office Hours: Mon. & Wed. 1:00-2:00, Tuesday 11:00-12:00
- Office: 214 MSB
- Phone: (810) 762-3423
- Email: farmerme@umflint.edu

Course Information:

- To date most of your classes have required you to write many lines of code. You have spent many late nights searching for syntax errors, etc. In other classes such as the algorithms class you have learned the detailed design of some common algorithms. For example, you have learned the cost in terms of memory and speed of many standard tasks, such as searching and sorting. In CS-382 we will step far back and learn how to develop engineering **models** of software systems. We will address the engineering **methods and techniques** associated with the key stages of engineering a software system. These models will ultimately make the coding of the system much easier. For very large systems coding would be impossible without this.
- In the second class, CS-383, we will address the aspects of **management and control** of the development of a software product. The objective of this second class is to provide the student with the knowledge of a variety of process models within which to apply the tools and techniques developed in 382. It also will provide the student with the skills to manage and document a software development.

Course Objectives:

- The goal of the class is to evolve you from programmers to software engineers. We will learn the modeling techniques needed to accomplish this. Our final products are graphical models and high level text descriptions. The objective is to provide the students with a good understanding and experience in using a set of 'tools' to methodically and professionally engineer a software product. In industry this skill is what separates a typical programmer from a potential project leader.
- Note, there will be **no code written** in this class. We are focusing on understanding how to analyze requirements and devise a design. The tools and methods to make this job easier are a key element of what will be learned in this class
- The *specific course objectives* are as follow:
 - Be able to clearly articulate a computer science-related project scope,
 - Be able to realistically estimate the effort of the task,

- Be able to develop a schedule for a set of tasks,
- Be able to perform the software engineering requirements analysis and describe them clearly
- Be able to design a software system and apply technologies from a number of computer science courses
- Be able to code and test a software system.

Class Format:

- The best way to learn Software Engineering is by **doing**. We will use lectures to explain techniques and methods we will be using. Then we will use on-going in-class case study to try our hand at the techniques. This is where class participation is critical (See Grading below). The best part of SW Engineering is there is no single right answer. The beauty of a design is truly in the eyes of the beholder. Students will try the techniques on their own through a collection of homework assignments. NOTE: All homeworks must be created electronically (i.e.. NO HAND-DRAWN DIAGRAMS) and turned in via Blackboard.

Assessment Mechanisms:

- Your understanding of the course objectives will be assessed by your writing and my reviewing the following deliverables:
 - Project Scope Document (1-page)
 - Abbreviated Project Plan (includes, scope, list of deliverables, assignments of responsibilities to the group members, and most importantly a schedule)
 - Abbreviated System Requirements Document (this should include a Context Diagram, possibly a collection of Use Cases and a brief introductory explanation of overall system requirements)
 - Developed System (this will be a prototype to demonstrate the requirements defined above were met at some level)
 - Project Write-up (a detailed write up that summarizes the scope, requirements, and the design details. This is *not* simply a listing of the source code written)
 - Project Presentation (This will include MS Power point slides reviewing the materials defined above, and will also include a demonstration of the system).

Prerequisites:

- CSC-175 and CSC-275 (Extensive programming experience and experience with Object Oriented Programming)

Book:

- *Software Engineering: A Practitioner's Approach, 6/e*, by Roger Pressman.

Other Good Reference Books:

- *Software Engineering: 7/e*, by Ian Sommerville
- *Introduction to Software Engineering*, by Panjab Jalote
- *Object Design*, by Becky Wirfs-Brock
- *Strategies for Specification of Real-time Systems*, by Hatley and Pirbah
- *UML Distilled*, by Scott Fowler

Tools:

- *TBD drawing tool*
- *Microsoft Word* – available everywhere

Grading:

- Homework (25%), Mid-term (25%), Project (25%), Final (25%)
- Class participation is an expected portion of the course work
- Scale:

A's	B's	C's	D's
97-100: A+	87-89: B+	75-79: C+	65-69.9: D+
92-96.9: A	82-86.9: B	72-74.9: C	62-64.9: D
90-91.9: A-	80-81.9: B-	70-71.9: C-	60-61.9: D-
			0-59: E

Attendance:

- Regular class attendance is essential. We will be working through examples and an ongoing case study to understand how to use the methods and tools. Note this class is now taught in the multi-media room so you may view the videos when you cannot come to class. If you want to attend class completely online please confirm with me first.

Late Homework Policy:

- Late assignments will be reduced by 10% per day. NOTHING will be accepted after the following week's class.

Academic Misconduct:

- Copying others work, plagiarizing external references without giving due credit, and cheating in exams are strictly forbidden. Please take pride in your own work, and feel free to ask me if you need help or assistance to maximize your learning.

Proposed Schedule: (Note this schedule is subject to change)

Week	Monday	Wednesday
Week 1 (1/7)	Overview & Introduction (Ch. 1)	Concepts and Principles of Software and Software Engineering (Ch. 1 & Ch. 5)
Week 2 (1/14)	Systems Engineering (Ch. 6)	Requirements Engineering: Elicitation (Ch. 7.1-7.5)
Week 3 (1/21)	<i>Martin Luther King Day (no class)</i>	Requirements Engineering: Elaboration (Ch. 7.6-7.9)
Week 4 (1/28)	Analysis Modeling: Structured Methods (Ch. 8.1, 8.2, & 8.6)	Analysis Modeling: CRC Cards (Ch. 8.4 and 8.7) & Project Proposals Due
Week 5 (2/4)	Analysis Modeling: Class models (Ch. 8.7)	Analysis Modeling: Behavior models (Ch. 8.8)
Week 6 (2/11)	Design Engineering: Overview (Ch. 9.1 & 9.2)	Design Engineering: OO-methods (Ch. 9.3 & 9.4)
Week 7 (2/18)	Review for Exam	Mid-term
Week 8 (2/25)	<i>Spring Break</i>	<i>Spring Break</i>
Week 9 (3/3)	Architectural Design (Ch. 10.1 & 10.2)	Architectural Design (Ch. 10.3 & 10.4)
Week 10 (3/10)	Architectural Design (Ch. 10.5 & 10.6)	Mid-term Project Reviews
Week 11 (3/17)	Component-level Design (Ch. 11)	Component-level Design (Ch. 11)
Week 12 (3/24)	Component-level Design (Ch. 11)	Component-level Design (Ch. 11)
Week 13 (3/31)	Software Testing Strategies (Ch. 13)	Software Testing Strategies (Ch. 13)
Week 14 (4/7)	Software Testing Techniques (Ch. 14)	Software Testing Techniques (Ch. 14)
Week 15 (4/14)	Project Presentations	Project Presentations
Week 16 (4/21)	Class Review	<i>No class</i>