

University of Wisconsin Oshkosh  
Computer Science Department

CS 341: Software Engineering I  
Fall 2008

----- Course Syllabus -----

**COURSE SCHEDULE**

		<b>Days</b>	<b>Time</b>	<b>Location</b>
<b>Section 001</b>	Lecture	T	1:20-2:50pm	HS 202
	Lab	TH	1:20-2:50pm	HS 101C

**INSTRUCTOR** Dr. Kathy (Kate) S. Faggiani  
Computer Science Department

EMAIL [faggianik@uwosh.edu](mailto:faggianik@uwosh.edu)

PHONE (920) 424-2069

IN-PERSON OFFICE HOURS\*(HS 218):

MWF 11:20am-12:40pm

TTH 12:50-1:20pm

Or use Google CHAT: [kfaggiani@gmail.com](mailto:kfaggiani@gmail.com) whenever I'm online.

\*Also available other times by appointment

**PREREQ** COMP SCI 271 and 251, both with a grade of C or better.

**COURSE DESCRIPTION**

This course will provide an in-depth study and analysis of at least one large scale software system. Students will analyze, design, and partially implement an extensive software project. Case studies will address major system concerns such as specification, classification, inter-relationships, validation, and evaluation. Other topics include the use of UML, prototyping, data flow diagrams and CASE tools. (Source: University of Wisconsin Oshkosh Undergraduate Bulletin, 2007-2009, p. 129.)

**PURPOSE OF THE COURSE**

Software engineering involves to the *practice* of computer science, as opposed to the *theory* of computer science. As a discipline, software engineering focuses on the "application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software." (Source: [http://en.wikipedia.org/wiki/Software\\_engineering](http://en.wikipedia.org/wiki/Software_engineering), extracted on September 1, 2008). While software engineering is based on a computer science foundation, computer science principles alone are insufficient to effectively provide a background for the development of large, complex, and critical software systems. While many of your prior programming courses involve programming "in the small" as you complete relatively small, well-bounded computer programs, software engineering exposes you to programming "in the large" – and provides essential preparation for your contribution to the profession in the real world. It also builds background knowledge for participation in, and development of research projects to explore the impact of existing and emerging software engineering methods and techniques. CS 341 course is a required course in the Computer Science Software

Engineering and Computer Information Systems emphases, and an elective in the Computer Science emphasis. It is also an elective course in the Computer Science minor and the Computer Science emphasis for Management Information Systems majors.

## **COURSE OVERVIEW**

Software engineering involves a full range of activities from initial inception of a software idea or definition of a problem, to installation and use of computer software by an individual or organization. Rather than reading about the process and discussing the possibilities in a purely academic setting, you will learn and follow a software engineering methodology as you complete the development steps for a software program for a real industry client. Your learning experience will consist of a combination of classroom lectures and discussions, case studies, meetings with industry engineers and software developers, individual assignments, and team project work that will be completed as a part of a software engineering effort. Your active involvement and participation in learning the concepts, tools, and methods of software engineering will be essential to successful completion of the course.

## **LETTER TO THE STUDENT**

Welcome to Software Engineering II! I hope you're looking forward to a challenging and rewarding semester. I've added this letter to the syllabus to share some additional perspectives on the course that you may find informative and useful.

This course will help you see how the technical concepts and skills you've learned in your other courses fit into the bigger picture of software development. This course is the first of a two semester course sequence, and the first semester provides a broad overview of software engineering methods and focuses on writing good, clean code in the agile development tradition, requirements analysis and definition, software architecture, modeling, and design skills. The course will include a wide variety of hands-on, active learning experiences. The active learning experiences will include a team contribution to an open source software development project. If you are contemplating a career in software development or perhaps graduate study in the computer field, I encourage you to explore taking Software Engineering II as an elective in the Spring 2008 semester. In the second semester, you may engage in a software development project with a real-world client, or conduct a research project into a relevant aspect of software engineering.

The U.S. Bureau of Labor Statistics indicates that computer software engineers, specifically those engaged in application development, are among the top 20 occupations with the highest projected employment increases for 2006-2016 (<http://www.bls.gov/oco/reprints/ocor006.pdf>). The job prospects for graduates with at a bachelor's degree in computer science and with practical work experience are excellent. As of August 2007, there were almost 30,000 employment opportunities nationwide for people who have varying degrees of experience in software engineering and development, and almost 300 of these jobs were in Wisconsin (<http://www.dice.com> search with keywords software engineer/software development retrieved on Monday, September 1, 2008 ). Average salaries for entry level software developers and software engineers vary widely depending on industry and geographical location, but a starting salaries in the upper-40s to low -50s or higher are fairly common for those with a bachelor's degree and some experience. Salary information for a variety of computer-related occupations can be found in the ComputerWorld 2007 Salary Survey at:

[http://www.computerworld.com/spring/salary-survey.htm?activeYear=2007&type=job\\_level&parameter=0&page=2](http://www.computerworld.com/spring/salary-survey.htm?activeYear=2007&type=job_level&parameter=0&page=2) .

In addition to providing potential employment opportunities, background in software engineering will give you a richer understanding of the context in which most computer-related activity occurs. The education and background needed by a software engineer extend well beyond strong technical skills and frequently include: strong interpersonal skills, good oral and written communication techniques, ability to contribute to a team effort, adaptability, project management skills, understanding of business principles and strategies, and the ability to see a problem from the perspective of a user/client rather than from a programmer's perspective. Please note that the development of these skills is more challenging than simply acquiring knowledge of specialized vocabulary and facts that are easily memorized from a book, powerpoint slides, or a lecture. Analysis, problem-solving, communication, team work, and project management are activities in which you must actively engage, thus your learning experience will extend beyond the traditional classroom. Please plan accordingly to devote time to the learning process in this course. One rule of thumb applied to most university level courses is 3 hours per week outside of class for EACH CREDIT of a course in which you are enrolled. Nine hours per week PLUS the scheduled three hours of class/lab time is a reasonable **minimum** expectation of time you'll need to devote to the course. You can expect this course to be much more challenging and more time consuming than other courses, but you can also expect to learn a great deal for your investment in time and effort.

I hope this brief letter provides you with additional background about the course. For more details, please read the attached syllabus and if I've left anything out don't be afraid to ask!

Also, if you're wondering what to call me since my last name can be a real mouthful, my previous students have called me by my first name/nickname Kate, Dr. F, Dr. Kate, or those with a bit of Italian in their background have used the full-on "Dr. Faggiani". I try not to be too formal, so feel free to use my first name or whatever you find most comfortable.

I look forward to working with you this semester!

Your Instructor and Software Engineering "Coach",  
Dr. Kathy (Kate) Faggiani

## **MATERIALS AND RESOURCES**

REQUIRED TEXTBOOK	Larman, Craig. (2005). Applying UML and Patterns. Third Edition. Prentice-Hall. ISBN: 0-13-148906-2.
OPTIONAL TEXTBOOK	Martin, Robert C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. ISBN: 0-13-235088-2. (Available at SafariOnline as an ebook).
SOFTWARE	NetBeans 6.1 – Available in all labs and free download at: <a href="http://java.sun.com/javase/downloads/netbeans.html">http://java.sun.com/javase/downloads/netbeans.html</a> . Select the <b>Java SE</b>

**Development Kit 6u7 and NetBeans IDE 6.1 Bundle** to download both the NetBeans development environment and JDK6.

OTHER

A minimum 1GB jump drive; you also have access to network storage on the campus server. Additional resources will be used on the internet and links will be provided in D2L.

### LEARNING OUTCOMES AND ASSESSMENT OF STUDENT LEARNING

Learning outcomes represent the knowledge, skills and abilities, and attitudes/behaviors you will acquire as a result of successful participation in this course. The learning objectives for Software Engineering I are listed below. Please note that each learner’s progress will be assessed at regular intervals using a variety of different assessment tools and techniques; in addition, learners will have the opportunity to self-assess their progress in some areas using D2L online activities.

**Table 1. CS 341 Learning Outcomes and Assessment**

<b>Course Outcome</b>	<b>Assessment Activity</b>
1. Describe the concepts and principles that guide software engineering practice.	Exam In-class activity
2. Explain the relationship between systems engineering and software engineering.	Exam In-class activity
3. Define the requirements engineering process and associated activities for both computer-based and web-based systems.	Exam Project deliverable
4. Define the analysis modeling process and associated activities for both computer-based and web-based systems.	Exam In-class activity Project deliverable
5. Define the design engineering process and associated activities for both computer-based and web-based systems.	Exam In-class activity Project deliverable
6. Describe testing strategies and techniques for computer-based and web-based systems.	Project deliverable In-class activity Project deliverable
7. Describe architectural design, styles, and patterns of computer-based and web-based systems.	Exam In-class activity
8. Define the guiding principles of good user interface design.	Exam In-class activity

9. Describe the use of component-level design in software engineering.	Exam In-class activity
10. Describe the project management concepts applied to software engineering projects.	Exam
11. Apply software engineering principles and methods to produce a solution to a significant problem.	Project deliverable
12. Select and apply appropriate analysis modeling practices and tools to a significant problem.	Exam Project deliverable
13. Select and apply appropriate requirements capture techniques and tools to a significant problem.	Exam Project deliverable
14. Select and apply appropriate design engineering techniques to a significant problem.	Exam Project deliverable In-class activity
15. Design and implement an effective test strategy and plan.	Project deliverable
16. Apply good user interface design principles to the development of a solution to a significant problem.	In-class activity Project deliverable
17. Identify appropriate situations for use of different software architectures.	Exam
18. Identify appropriate situations for use of components-based design.	Exam
19. Establish effective communication plans with clients and co-developers.	Review of project progress Observation
20. Conduct effective presentations for clients.	Observation
21. Conduct effective design reviews with co-developers.	Observation Review by client
22. Develop and adhere to project plans and schedules.	Review of project progress
23. Seek timely assistance from others when individual efforts to resolve problems have been unsuccessful.	Observation
24. Establish an atmosphere of trust and mutual commitment with clients.	Review by client

The learning outcomes described in Table 1 relate directly to the Computer Science Program Objectives found at: [http://www.uwosh.edu/departments/computer\\_science/cs\\_major/program\\_objectives.php](http://www.uwosh.edu/departments/computer_science/cs_major/program_objectives.php)

If you have a major or minor other than Computer Science, please consult the learning objectives for your specific program to determine the contribution of CS 341 learning outcomes to your degree program.

### DETERMINATION OF GRADES

The previous section on learning outcomes and assessment of student learning described what you can expect to learn in the course and how it will be assessed. The Course Policies section described the importance of meeting due dates, and the different elements that will contribute to your overall grade. The items that will contribute to your final grade and the percent of contribution are listed below:

	<u>Points Possible</u>	
Open Source/Software Engineering Project	40%	600
In-Class Activities	10%	100
Exams	40%	300
Lab Quizzes/Lab Activities	<u>10%</u>	<u>100</u>
	100%	1000

Your letter grade for the course will be determined based on total points earned in the course as follows:

Point Range	Grade	Point Range	Grade
920 to 1000	A	720 to 789	C
890 to 919	AB	690 to 719	CD
820 to 889	B	600 to 689	D
790 to 819	BC	0 to 599	F

### COURSE POLICIES

Please read the follow course policies carefully and make sure you email [faggianik@uwosh.edu](mailto:faggianik@uwosh.edu) or [kfaggiani@gmail.com](mailto:kfaggiani@gmail.com) with any questions you have. You will be responsible for conforming to all course policies, related rewards, and consequences as they are documented here, whether or not they are explicitly discussed in class.

### ATTENDANCE and IN CLASS ACTIVITIES

*“Eighty percent of success is showing up.”*

*-Woody Allen*

It is my goal to make the scheduled meetings so interesting that you don’t want to miss one! Since the course involves much more than memorizing information presented on Powerpoint presentations or in readings, or writing code, your attendance to participate in classroom activities is vital to your success in the course. Each of you make a unique contribution to the classroom environment through your shared knowledge and experience – if you’re not in class, the class is less than it can be and everyone suffers. Please plan to attend each session. If you miss a class in which an activity is conducted for which points are assigned to participants, you will not be able to make up the points unless the

instructor has approved your absence in advance of the absence. The total of all class activity points will contribute to 10% of your course grade.

## PREPARATION

*“To be prepared is half the victory.”*

- Miguel de Cervantes Saavedra

In addition to attending scheduled sessions, the successful learner must be prepared. The D2L course site lists reading projects, required and optional learning activities, project instructions, due dates, and other learning tools. It is the responsibility of the learner to complete scheduled items prior to the class period and be prepared to engage in active learning exercises. Please note that if you are prepared for class, it will be much easier to earn the maximum possible points on in-class activities.

As stated in my “Letter to the Student”, you can reasonably expect to spend three hours outside of class for each 1 credit of class – meaning 9 hours per week *in addition to* class/lab attendance. When your project kicks into gear, you may need to spend more time than this estimate. Please plan to be prepared for each class session by completing the required activities.

## DUE DATES

*“A perfect method for adding drama to life is to wait until the deadline looms large.”*

-Unknown

Although I will be providing you with due dates for many class activities, on occasion you will be required to specify a time schedule for some course activities. You must commit to meeting all due dates and deadlines as stated both by the instructor and per your own plan. If you fail to meet a stated due date or deadline, you must commit to a definite date/time by which the item will be completed in writing to the instructor (email is sufficient) prior to the missed date/deadline. If you fail to comply with this requirement, you will receive no points for the missed item.

One of the unique aspects of this class, compared to some of your other Computer Science courses, is that you will be asked to repeat unsatisfactory work – meaning if you do the work to a high standard of quality the first time, by the due date, little re-work will need to occur. If work submitted by the due date is incomplete or of inferior quality, you will be creating more work for yourself. Since many of the activities you will be completing are based on prior steps, falling behind has a major impact on your learning outcome in the course. Please plan to meet all deadlines and due dates as stated, or email me a date/time by which a late item will be submitted. If you miss a due date without prior discussion and approval of the instructor, you will receive no points for the assigned item.

Also note that I have numerous office hours during the week, including virtual office hours, and I can be reached by email. I am happy to help you succeed in the course so don't be afraid to ask for assistance when needed. Please don't let lack of understanding stand in the way of your success.

## PROJECTS

*“If your project doesn't work, look for the part that you didn't think was important”*

-Arthur Bloch

Software engineering is a complex process, and it's extremely difficult (if not impossible) to teach it well in a classroom environment. Even the best textbooks, case studies, or simulations fall far short of

preparing a learner for participation in the actual process of software engineering. Thus this course will require, as the primary learning experience, an open source software engineering project that will allow you to make a meaningful contribution to an on-going software development project. Students will be organized into teams, and each team will be assigned to a project. The project work will provide students the opportunity to apply, in a very real sense, the software engineering process, methods, and tools learned in classroom activities to an actual development project. The project will comprise approximately 40% of your overall course grade.

#### EXAMS

*“Examinations are formidable even to the best prepared, for the greatest fool may ask more than the wisest [person] can answer.”*

*-Charles Caleb Colton*

In addition to demonstrating skills/abilities and professional attitudes/behavior in the course, each student must demonstrate an understanding of key principles of software engineering, and modeling/design skills. These aspects of the learning process will be assessed in two exams, each worth 150 points. Exams will contribute 40% of the overall course grade.

All exams will use the D2L Quiz platform or a combination of media. A sample exam will be available for students to practice with prior to the first exam. This will allow students to gain familiarity with the testing platform and know what to expect during the exam.

#### LAB/LAB QUIZZES

*“The work of the individual still remains the spark that moves mankind ahead even more than teamwork.”*

*-Igor Sikorsky*

The class meets twice each week – the first meeting covers content and in-class activities. The second meeting is a hands-on learning experience in the HS101C computer lab. The purpose of the lab sessions is to provide you with modeling time and practice with the methods and tools of software engineering. In addition, each scheduled lab will begin with a brief quiz that covers the assigned reading and class content covered during the week. The points awarded on Labs/Lab Quizzes will constitute 10% of the overall course grade.

#### ACADEMIC HONESTY

*“Integrity is doing the right thing, even if nobody is watching”*

*- Unknown*

As an adult, you have the choice to complete the course requirements to the best of your abilities or sacrifice your integrity and reputation for what you may perceive as a necessity at the time. If, at any point during the semester, you feel pressured to commit an act of academic misconduct in order to successfully fulfill course requirements, please see me immediately. I’m sure we can come up with a plan of action that will help you succeed in the course and maintain your integrity. Please note that the following actions constitute academic misconduct and are subject to disciplinary action under the [UW Oshkosh Student Discipline Code](#) (2007): claiming the work of others as one’s own – whether the work is given willingly or unwillingly/unknowingly by another student, copied from an internet site of any kind contrary to course requirements, or originating anywhere other than as your own work product; cheating on an examination or gaining unauthorized access to examination materials; collaborating on work with others, contrary to the stated requirements of the course; failing to

appropriately identify and cite the sources or ideas presented in a work product for the course; submitting work previously presented in another course; tampering with or destroying work completed by other students; or assisting other students in any of these acts. Students who feel compelled to engage in academic misconduct will be subject to the penalties defined in UWS Chapter 14 of the UWO Student Discipline Code.

## COURSE CALENDAR

The table below is provided as a general timetable for topics and activities. Note that depending on progress in the course, this schedule may be modified. All modifications will be announced in class and posted as a news item on the D2L course site. Please see the Content section of the D2L course site for detailed information about weekly course activities, labs, projects, and related due dates.

Dates	Topics
Sept 3-5	Introduction to the Course <b>Lab 1/Quiz – Part A</b>
Sept 8-12	Team Skills Overview and Assessment Introduction to Systems Thinking Software Engineering, and Professional Practice Software Engineering Methodologies and Processes Clean Coding <b>Lab 1/Quiz – Part B</b>
Sept 15-19	Object-Oriented Analysis and Design Overview of Iterative Development and the Agile Unified Process Role of Process, Modeling, and Documentation <b>Lab 2/Quiz</b>
Sept 22-26	Software Development Project Planning and Management Requirements Analysis and Modeling: Use Cases and Domain Models <b>Lab 3/Quiz</b>
Sept 29 – Oct 3	Modeling System Interactions: Sequence and Collaboration Models Data Modeling and Database Design <b>Lab 4/Quiz</b>
Oct 6-10	Developing Effective Software Development Teams Prepare for Exam 1 <b>Exam 1 at 1:20pm on Oct 9 in HS101C</b>
Oct 13-17	Begin Team Project User Interface Design and Prototyping Modeling System Designs: Use Case Realizations, Patterns, Visibility, and Mapping to Code Test Driven Development in Practice <b>Lab 5/Quiz</b>
Oct 20-24	Modeling Generalization and Specialization: Domain Model Refinement Designing and Modeling System Behavior: Statechart Diagrams

Dates	Topics
	<b>Lab 6/Quiz</b>
Oct 27-31	Software Architecture: Design and Analysis Designing Persistence Frameworks with Patterns <b>Lab 7/Quiz</b>
Nov 3-7	Testing Strategies for Iterative Development Software <b>Lab 8/Quiz</b>
Nov 10-14	Professional Interaction between Software Engineers and Clients Prepare for Exam 2 <b>Exam 2 at 1:20pm on Nov 13 in HS101C</b>
Nov 17-21	Presentation Planning and Presentation Skills Team Project
Nov 24-30	Team Project <b>Thanksgiving Break – No Class on Nov 26, 27, 28</b>
Dec 1-5	Team Project
Dec 8-12	Team Project Deliverable Due at 1:15pm on Dec 9 <b>Team Project Presentations Due on Dec 11</b>

This course syllabus was developed based on the principles defined in “*The Course Syllabus: A Learning-Centered Approach*” by J. Grunert (1997).