

**University of Wisconsin Oshkosh
Computer Science Department**

**COMP SCI 342: Software Engineering II
Spring 2008**

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

COURSE SCHEDULE

| | | Day | Time | Location |
|--------------------|-------------|------------|-------------|-----------------|
| Section 001 | Lecture/Lab | TTH | 4:40-6:10pm | HS208/HS101C |

INSTRUCTOR Dr. Kathy (Kate) S. Faggiani
Computer Science Department
EMAIL faggianik@uwoshkosh.edu PHONE (920) 424-2069
IN-PERSON (HS 218) : TTH 11:30am-12:30pm; MW 8:00-11:00am
Or Click on LIVE CHAT at <http://www.professorfaggiani/kfaggiani> (only available during posted office hours).

PREREQ COMP SCI 341 with a grade of C or better.

COURSE DESCRIPTION

Software engineering is the second of a two-semester sequence on the topic of modern Software Engineering tools and techniques. Topics covered include design patterns, the Unified Modeling Language (UML), component-based software development, advanced OO design and analysis, refactoring, and other techniques such as extreme programming. An extensive software development project will allow for practical application of the discussed techniques. (Source: University of Wisconsin Oshkosh Undergraduate Bulletin, 2007-2009, p. 129.)

PURPOSE OF THE COURSE

Software engineering blends the technical concepts and skills learned in previous courses with systematic approaches to the effective and efficient development of computer software. This course is a required course in the Computer Science Software Engineering emphasis, and an elective in other Computer Science emphases and programs.

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 your experience in Software Engineering I has helped you develop a good foundation for the second semester course in software engineering. As you learned last semester, software engineering encompasses a broad set of knowledge and skills beyond coding, which included: working with a client to analyze and define software requirements, using different modeling approaches and perspectives, capturing and tracking requirements, communicating effectively with clients in both oral and written formats, and working well as a part of a team.

In this course, you will continue to improve and apply many of the techniques and skills discussed in the first semester. In addition, you will learn new methods and techniques that you will apply to the on-going project you began last semester. Similar to last semester, the development of the knowledge and skills needed in software engineering is more challenging than simply acquiring knowledge of specialized vocabulary and facts that are easily memorized from instructor notes, books, powerpoint slides, or a lecture. Analysis, problem-solving, communication, team work, and computer programming 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! Do review the syllabus section on learning objectives for a clear view of what you can expect to learn this semester.

Thanks to the thoughtful comments and feedback you provided on last semester's course, you'll see some changes this semester. As we move forward, please feel free to let me know if there is anything I can do to further enhance your learning experience in the course.

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

MATERIALS AND RESOURCES

| | |
|-------------------|---|
| REQUIRED TEXTBOOK | Pressman, Roger S. (2005). Software Engineering A Practitioner Approach. 6 th Edition. McGraw-Hill Series in Computer Sciences. ISBN: 0-07-285318-2 |
| SOFTWARE | MagicDraw UML modeling software or Visio, available in campus labs. Visio UML is also available for free download via the MSDN Academic Alliance program. See Wing Huen for details. Other software as required - TBA |
| OTHER | A minimum 512M to 1GB jump drive; you also have access to network storage on the campus server via the Q drive, or via the internet with TitanFiles. Additional resources will be used on the internet and links will be provided in D2L. |

ACCESS TO COURSE SYLLABUS AND D2L COURSE SITE

D2L will be used extensively in the course as a repository for all course materials and for course enhancement activities. The course syllabus, course schedule, weekly activity plans, projects, and all related course information are available to students enrolled in CS 342. In addition, students may be asked to submit course deliverables to the appropriate digital dropbox in D2L by the designated due dates, and will take quizzes/exams online in D2L as directed. On occasion, students may be asked to participate in online discussions, online chat sessions, and other online activities which will supplement and enhance the scheduled face-to-face class periods.

To access D2L, click on the D2L Desire 2 Learn link on the right side of the UWO home page at www.uwosh.edu and log in with your UWO email username and password. Your default username for Titan Mail and D2L is: first 5 letters of your last name + first letter of your first name + last 2 digits of your student ID number, such as **faggik11** and your default password is your 7-digit Campus ID number from your Titan card.

After you have logged in to D2L, click the CS 342 course link from the list of courses toward the bottom of your D2L home page. Then click Content on the CS 342 course home page's menu bar and locate Course Syllabus in the Course Documents module. You can click on the syllabus link to open and print a hard copy of the syllabus.

LEARNING OBJECTIVES

Learning objectives 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 II are provided below. Please note that each learner's progress will be assessed at regular intervals using a variety of different assessment tools and techniques.

Table 1. Comp Sci 342 Learning Objectives

| |
|--|
| 1. Given a project narrative and set of requirements, the student will be able to <i>create</i> UML design models which include: a domain object model and appropriate class diagram iterations. |
| 2. The student will be able to <i>recall</i> and <i>identify</i> different object-oriented software architectures. |
| 3. The student will be able to <i>identify</i> design patterns by <i>analyzing</i> existing UML class diagrams. |
| 4. The student will be able to <i>implement</i> the assigned design pattern in programming code. |
| 5. The student will be able to <i>combine</i> design and implementation classes into packages and components that are developed as a part of a project. |
| 6. The student will be able to <i>develop</i> a task set, project schedule, and project tracking plan for a given software project. |
| 7. The student will be able to <i>recall</i> and <i>describe</i> the key characteristics of agile software development. |
| 8. The student will be able to <i>recall</i> and <i>describe</i> the key elements of test-driven software development. |
| 9. The student will be able to <i>distinguish</i> among different software testing strategies, and identify an appropriate software testing strategy for a given project. |
| 10. The student will be able to <i>apply</i> a variety of software engineering methods, techniques, and tools to the development of a relatively complex software development effort. |
| 11. The student will be able to <i>describe</i> software quality measures and metrics. |
| 12. The student will be able to <i>apply</i> software quality tools and metrics to <i>evaluate</i> of a given software system. |

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

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 342 learning outcomes to your degree program.

DETERMINATION OF GRADES

The previous section on learning outcomes described what you can expect to learn in the course. 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> | |
|-------------------------------|------------------------|------------|
| Software Engineering Project: | | |
| Client Evaluation | 25% | 250 |
| Instructor Evaluation | 25% | 250 |
| Exams (2 @150 points) | 30% | 300 |
| Assignments (5@40 points) | <u>20%</u> | <u>200</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 or 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

“Eighty percent of success is showing up.”

-Woody Allen

Class sessions will meet both Tuesday and Thursday. Many of the Thursday sessions will be structured lab time for you to work on assignments, or for your team to work on key project deliverables. In addition, your project team will schedule a two hour weekly meeting time. You need to plan to attend all scheduled sessions. It is my goal to make the scheduled class/lab meetings so interesting that you don't want to miss one! However, that may not always be the case. Since the course involves much more than memorizing information presented in instructor notes, on Powerpoint presentations, or in readings, 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.

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.

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 different aspects of this class 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.

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.

PROJECT

“For every complex problem, there is a solution that is simple, neat, and wrong.”

Charles Caleb Colton

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 include the continuation of last semester's project as the primary learning experience. Student teams will continue, with some modification, from last semester's projects. 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/lab activities to an actual development project.

Performance and learning assessment on the project will be evaluated from two perspectives: the client and the instructor. Client and instructor will evaluate the performance of a team and the scores will be combined to arrive at a final project score for the student. This score will comprise 50% of the overall grade in the course. Evaluation rubrics for both client and instructor will be provided to the student by Week 3 of the semester.

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. These aspects of the learning process will be assessed in two exams, each worth 150 points. Exams will contribute 30% of the overall course grade.

All exams may use a combination of paper/pencil and the D2L Quiz platform. 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. Exams will be given on Thursday, and graded exams will be returned the following Tuesday. All scores will be posted on D2L as they become available.

ASSIGNMENTS

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

-Igor Sikorsky

In addition to the course project, there will be several assignments that each student will complete individually. The purpose of these assignments will be modeling and practice with the methods and tools of software engineering. The points awarded on individual assignments will constitute 20% of the overall course grade. All assignments will be due on Monday following the week in which they are assigned, and results will be returned to students on Tuesday, the day following the due date.

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. For complete information about reading assignments, labs, projects, exams, handouts, and all due dates, please refer to the information posted in the Content section of the D2L course site. 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 | Week | Topics |
|--------------|-------------|---|
| Feb 4-8 | Week 1 | Comp Sci 341 De-Briefing and Performance Awards Introduction to Course Agile Development/Teamwork and Agile Teams Project Management and Software Project Scheduling |
| Feb 11-15 | Week 2 | Agile Modeling: User Interface Prototyping |
| Feb 18-22 | Week 3 | Agile Modeling: Class Diagrams Software Design Example: From Object Models to Implementation Class Diagrams Software Design and Architecture |
| Feb 25-29 | Week 4 | From Class Diagrams to Code: Identifying Design Patterns and Implementing Patterns in Code |
| Mar 3-7 | Week 5 | Proving Models in Code Exam I – covers content from Weeks 1-5 |
| Mar 10-14 | Week 6 | Agile Coding: Extreme Programming and Test-First Development Role of Use Cases in Test-First Development Unit and Integration Testing with Java and PHP |
| Mar 17-21 | Week 7 | Software Testing Strategies and Techniques More on Unit and Integration Testing with Java and PHP |
| Mar 24-28 | Week 8 | SPRING BREAK |

| | | |
|----------------|---------|--|
| Mar 31 – Apr 4 | Week 9 | Software Quality Management and Metrics |
| Apr 7-11 | Week 10 | Quality in Agile Development Exam II – covers content from Weeks 1-10 |
| Apr 14-18 | Week 11 | Software Engineering Project |
| Apr 21-25 | Week 12 | Software Engineering Project |
| Apr 28-May 2 | Week 13 | Software Engineering Project |
| May 5-8 | Week 14 | Project Presentations |
| May 12-15 | Week 15 | Evaluation and Project De-Briefing |