

# Software Engineering I

## CS 341 - Fall 2016

**Instructor:** George Thomas                      **Office:** Halsey 218  
**Email:** [thomasg@uwosh.edu](mailto:thomasg@uwosh.edu)                      **Phone:** 424-2069

**Office Hours:** MW        1:30-3:00,  
                         TR        10:30-11:30,  
                         Or by appointment

**Lectures:**        W        3:30-5:00        HS 237  
**Labs:**            M        3:30-5:00        HS 101C

**Credits:** This is a 3 credit course.

**Prerequisites:** CS 262 with a grade of C or better *and* junior-level standing, OR CS 271 with a grade of C or better.

### Textbook:

#### Required:

1. *Clean Code: A Handbook of Agile Software Craftsmanship*, 1st Edition, by Robert C. Martin, Prentice Hall
2. *The Clean Coder: A Code of Conduct for Professional Programmers*, 1st Edition, by Robert C. Martin, Prentice Hall

#### Recommended:

1. *Essentials of Software Engineering*, 3rd Edition, by Frank Tsui, Orlando Karam and Barbara Bernal, Jones & Bartlett Learning
2. *Engineering Software as a Service: An Agile Approach Using Cloud Computing*, 1<sup>st</sup> Edition, by Armando Fox and David Patterson, Strawberry Canyon LLC

**Course Website:** UWO D2L (<http://www.uwosh.edu/D2L>)

**Note: If you have special needs, please come and talk to me at the end of the first class.**

### Course Overview

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, etc.

The specific learning objectives are listed below.

1. Describe the concepts and principles that guide software engineering practice.
2. Explain the relationship between systems engineering and software engineering.

3. Define the requirements engineering process and associated activities for both computer based and web-based systems.
4. Define the analysis modeling process and associated activities for both computer-based and web-based systems.
5. Define the design engineering process and associated activities for both computer-based and web-based systems.
6. Describe testing strategies and techniques for computer-based and web-based systems.
7. Describe architectural design, styles, and patterns of computer-based and web-based systems.
8. Define the guiding principles of good user interface design.
9. Describe the use of component-level design in software engineering.
10. Describe the project management concepts applied to software engineering projects.
11. Apply software engineering principles and methods to produce a solution to a significant problem.
12. Select and apply appropriate analysis modeling practices and tools to a significant problem.
13. Select and apply appropriate requirements capture techniques and tools to a significant problem.
14. Select and apply appropriate design engineering techniques to a significant problem.
15. Design and implement an effective test strategy and plan.
16. Apply good user interface design principles to the development of a solution to a significant problem.
17. Identify appropriate situations for use of different software architectures.
18. Identify appropriate situations for use of components-based design.
19. Establish effective communication plans with clients and co-developers.
20. Conduct effective presentations for clients.
21. Conduct effective design reviews with co-developers.
22. Develop and adhere to project plans and schedules.
23. Seek timely assistance from others when individual efforts to resolve problems have been unsuccessful.
24. Establish an atmosphere of trust and mutual commitment with clients.

**Course Grading Policy:** Your grade for this course will be based on four components, namely exams, lab exercises, readings and quizzes, and a team-based software engineering project.

Your overall numerical grade for the course will be computed as the weighted sum of the component grades using the following weights:

<b>Component</b>	<b>Weight</b>
Exams (2)	20%
Labs	25%

Readings and Participation	15%
Team Project	40%

**Each component will have specific grading criteria and policies that will be described later.**

Your letter grade for the course will be computed as follows:

Numerical Score	Grade	Numerical Score	Grade
>=92	A	72-78	C
90-92	A-	70-72	C-
88-90	B+	68-70	D+
82-88	B	62-68	D
80-82	B-	60-62	D-
78-80	C+	<60	F

While this overall grading scheme is fixed, I will be happy to discuss any issue you may have with individual grades. If you notice a mistake or have a question regarding a specific grade, please come and talk to me *as soon as possible*. Do not wait until the end of the semester to bring up grading issues.

If you miss a scheduled exam, you **may** be able to take a make-up exam provided you give the instructor a valid justification (in the form of a written excuse from a medical doctor or the Dean of Students Office) ahead of time if possible.

Tentative Exam Dates are as follows:

- **Exam 1 - Wednesday, 10/19**
- **Exam 2 - Wednesday, 12/14**

**Team Project:** Software engineering is a complex process, and it is impossible to teach it well strictly in a classroom environment. Even the best textbooks, case studies, or simulations fall short of preparing a learner for participation in the actual process of software engineering. Thus, as mentioned previously, this course will require the successful completion of a significant software engineering project as one of the components of your grade. Students will be organized into teams, and each team will choose a project, subject to some restrictions. The project will comprise a significant chunk of your overall course grade, distributed among the different phases of your project. Each phase will have deliverables and deadlines.

You must commit to meeting all due dates and deadlines as stated both by the instructor and per your own team's 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 prior to the missed date/deadline. If you fail to comply with this requirement, you will receive no points for the missed item.

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.

**Calendar:** Please see the course site for detailed information about weekly course activities, labs, projects, and related due dates.

**Academic Honesty:** Unless otherwise stated, all work must be that of a single student (the one whose name appears on the submission, that is). You may not “borrow” any piece of code or design of any length from someone else, unless you can live with a zero and the other potential academic sanctions of cheating (see [UWO Student Discipline Code 2007](#), Chapter UWS 14).

**Attendance and Participation:** You are expected to not only attend **every** class meeting but also to come **prepared** for and **participate** actively in it. I **strongly encourage you to ask any question** or raise any issue you have with the course either during or at the end of class, or during my office hours. I will also gladly meet with you by appointment.