CS 460 Software Engineering I

Fall 2017, 4 Credits

Contact Information

Instructor:	Scot Morse
Email:	morses@wou.edu
Website:	http://www.wou.edu/~morses
Office Phone:	(503)-838-8921
Building/Room:	ITC 301
Class Meeting Times:	08:00 – 08:50 MTWR
Office Hours:	See Class Website for Hours
Textbook:	

Required: *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*, by Scott W. Ambler and Mark Lines, IBM Press 2012

Optional: *Pro ASP.NET MVC 5*, by Adam Freeman, Apress 2013

Recommended: *C# 6.0 in a Nutshell: The Definitive Reference*, by Joseph Albahari and Ben Albahari, O'Reilly 2016

_____

Course Description & Prerequisites

This course covers an introduction to the fundamental principles and practices of software engineering. Covers both theory and practical aspects of the first activities of large, group-developed software projects, including tools and environments, software requirements engineering, introduction to software architecture, design and modeling and modern software processes. Students will learn the specific languages, tools, environments and processes necessary to complete the remainder of the capstone course. Computer Science majors must complete the required senior capstone courses in sequence: CS 460 then CS 461 and finally CS 462.

Prerequisites: CS 361 (Algorithms) and CS 364 (Information Management)

Course Outcomes/Objectives

After completing this course, students will be able to
1. understand the uses of and demonstrate the capability to effectively employ tools and techniques used in support of professional software development.
2. describe functional requirements using common methods (e.g. use cases and user stories) and then create appropriate models to implement the required functionality.
3. create a small cloud-hosted database driven web application using a professional environment that utilizes modern API's, technologies and standards.

See Class Webpage

TOPICS (BY ACM KNOWLEDGE AREA)

SDF/Development Methods
- Program correctness, role and use of contracts, including pre-and post-conditions
- Simple refactoring
- Modern Programming environments, using library components and their API's

SE/Tools and Environments
- Software configuration management and version control
- Requirements analysis and design modeling tools
- Programming environments
- Tool integration

SE/Requirements Engineering

- Functional Requirements, e.g. use cases or user stories
- Properties of requirements
- Describing system data, class diagrams or E-R diagrams
- Non-functional requirements
- Use of requirements
- Specification

SE/Software Design

- Design principles
- Design paradigms, OOP, event driven, data-centric, MVC
- Design patterns and refactoring
- Introduction to software architecture concepts and standard architectures
- Use of components
- Application frameworks

SE/Software Construction

- Coding practices, standards

SE/Software Processes

- Introduction to software process models

- Agile and Scrum methodologies, activities

HCI/Programming Interactive Systems

- Software architecture patterns, MVC
- Modern libraries/environments
- Declarative interface specification, stylesheets, DOM
- Data-driven applications (database-backed web pages)

PBD/Web Platforms

- Web programming languages
- Web platform constraints
- Software as a Service (SaaS)
- Web standards

Other

- Website API programming, both as consumer and producer
- Cloud systems and services, provisioning, management, operations

COURSE REQUIREMENTS

A student's course grade is determined by scores on components in the following categories:

Programming and Homework Assignments; Portfolio	50%
Midterm Exam	15%
Concepts Final Exam	15%
Hands-on Programming Final Exam	15%
Participation and Attendance	5%

NOTE: The programming final exam is a proficiency exam covering the languages, environments, services and tools required for the entire capstone sequence. Students must pass this exam with a minimum proficiency in order to pass the class and continue on with CS 461. Students not passing the exam will receive a C- or below, regardless of their grade calculated from the other course requirements.

GRADING SCALE

COMPONENTS				
	A	92-100%	C	72-77%
	A-	91-92%	C-	70-71%
	B+	88-89%	D+	68-69%
	B	82-87%	D	62-67%
	B-	80-81%	D-	60-61%
	C+	78-79%	F	0-59%

Academic Honesty Policy

Code of Student Responsibility 574-031-0030 Specific Standards and Policies

The following list of prohibited forms of conduct is not all inclusive since it is not possible to list all potential   violations. The University requires that all students behave in a manner congruent with established community standards and in a manner conductive to the development of the individual. Actions detrimental to the mission of the University and the legitimate activities of the academic community which constitute the University are in violation of this Code and may be subject to judicial procedures.

*Academic dishonesty, which includes but is not limited to:

Cheating: intentional use or attempted use of artifice, deception, fraud, and/or misrepresentations of one's academic work;

Fabrication: unauthorized falsification and/or invention of any information of citation in any academic exercise;

Facilitating dishonesty: helping or attempting to help another person commit an act of academic dishonesty. This includes students who substitute for other persons in examinations or represent as their own papers, reports, or any other academic work of others;

Plagiarism: representing without giving credit the words, data, or ideas of another person as one's own work in any academic exercise. This includes submitting, in whole or in part, prewritten term papers of another of research of another, including but not limited product of commercial vendor who sell or distribute such materials. And the appropriation of and/or use of electronic data of another person or persons as one's own, or using such data without giving proper credit for it; or

Any use or attempted use of electronic devices in gaining an illegal advantage in academic work in which use of these devices is prohibited, and such devices include but are not limited to cell phones, pdas, laptops, programmable calculators, etc.

Programming language source code is no different where academic honesty is considered. The code you write in a Computer Science course is not essentially different from the paper you write for a Literature or History class. They are both your own work and ideas.

Disability Statement

If you have a documented disability that requires academic accommodations at Western Oregon University, you are required to have your accommodations coordinated through the Office of Disability Services (ODS). ODS is located in APSC, Rm. 405. Phone: 503-838-8250 VTTY. Email: ODS@wou.edu

Veterans and active duty military personnel with special circumstances are welcome and encouraged to communicate these, in advance if possible, to the instructor.

Wolf Connection System Referral Program

Students in this class may be referred to the WOU Student Success Specialist (SSS) if the instructor determines their performance in the class is placing them at academic risk. The SSS will offer to work with referred students to address issues and develop a student success strategy. Irrespective of whether

A REFERRAL HAS OR HAS NOT BEEN MADE, YOU ARE ULTIMATELY RESPONSIBLE FOR TRACKING YOUR OWN PROGRESS IN THIS COURSE.