CS 320
Fundamentals of Software Engineering

Lecture 1: Overview
Introduction

• Instructor

• Xinghui Zhao

• Email: x.zhao@wsu.edu

• Phone: 360-546-9110

• Office: VECS 201X

• Office Hours: When door is open or by appointment
CS 320

Location: VSCI 12

Time: T, Th 1:25pm - 2:40pm

Webpage:

- [http://dsr.encs.vancouver.wsu.edu/Teaching/320/](http://dsr.encs.vancouver.wsu.edu/Teaching/320/)

- ANGEL for submitting assignments ([http://lms.wsu.edu](http://lms.wsu.edu))
Big Picture

- Software Engineering Courses
  - CS320 (this one)
  - CS420/421 (Brand new capstone course)
    - CS420 Software Design Project I (Fall 2015) - taught by Dr. Zhao
    - CS421 Software Design Project II (Spring 2016) - taught by Dr. Cochran
Class Rules

- Cell phones and beepers should be turned off
- Laptop is allowed, but no facebook, no twitters
- Promptness: class starts at 1:25pm sharp
- Coming late is fine as long as you don’t disturb the class
Course Text

✧ Required text:

✧ Software Engineering (9th Edition), Ian Sommerville, Addison Wesley Inc.

✧ Reference Materials:

✧ UML Distilled (3rd Edition), Martin Fowler, Addison Wesley Inc.

✧ The Mythical Man-Month, Frederick P. Brooks, Addison Wesley Inc.
Eric Klinginsmith

eric.klinginsmith@email.wsu.edu
Prerequisites

- Prerequisites
  - CS 224 Programming Tools
  - Math 216 Discrete Structures
  - Engl 402 Technical Writing (concurrent enrollment allowed)

- Prerequisite Topics
  - Object-oriented programming languages
  - Principles of technical writing
  - Use of UNIX or Windows environment for coding, compilation, debugging, and testing
Topics

• Object-oriented design and programming
• Software requirements and specification
• Software engineering processes
• Test and debugging
Evaluation

* Assignments and project 50%
* Midterm 20%
* Final Exam 25%
* Discretionary 5%

<table>
<thead>
<tr>
<th>Grade</th>
<th>Range</th>
<th>Grade</th>
<th>Range</th>
<th>Grade</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>94 - 100%</td>
<td>A-</td>
<td>91 - 93 %</td>
<td>B+</td>
<td>87 - 90%</td>
</tr>
<tr>
<td>B-</td>
<td>80 - 82%</td>
<td>C+</td>
<td>77 - 79%</td>
<td>C</td>
<td>73 - 76%</td>
</tr>
<tr>
<td>D+</td>
<td>68 - 70%</td>
<td>D</td>
<td>62 - 67%</td>
<td>D-</td>
<td>60 - 61%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td>83 - 86%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C-</td>
<td>71 - 72%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
<td>≤ 59%</td>
</tr>
</tbody>
</table>
Academic Integrity

- Do not cheat
- To avoid plagiarism
  - Put pens away when discussing problem with other people
  - Acknowledge any help you received in your assignments: state name of person
- Write your own code
What is Software

- Software is

  - instructions (computer programs) that when executed provide desired features, function, and performance
  - data structures that enable the programs to adequately manipulate information
  - documentation that describes the operation and use of the programs
What is Software?

- Software is developed or engineered, it is not manufactured in the classical sense
- Software doesn’t “wear out”
- Although the industry is moving toward component-based construction, most software continues to be custom-built
Wear Vs. Deterioration

Q: What about hardware?
What is Software Engineering?

* Software engineering is concerned with theories, methods, and tools for the development of quality software to help solve problems
Why SE is important?

- The economies of all developed nations are dependent on software
- More and more systems are software controlled
- Software engineering expenditure represents a significant fraction of GNP in all developed countries
How can things go wrong?

**THE LIFE OF A SOFTWARE ENGINEER.**

Clean slate. Solid foundations. This time I will build things the right way.

**MUCH LATER...**

Oh my. I’ve done it again, haven’t I?
How can things go wrong?
What is the difference between software engineering and computer science

* Computer science is concerned with theory and fundamentals

* Software engineering is concerned with using computer theory, languages etc. to implement a solution to a problem
A Layered Technology

Software Engineering
The Essence of Practice

- Understand the problem
  - Communication and analysis
- Plan a solution
  - Modeling and software design
- Carry out the plan
  - Code generation
- Examine the result for accuracy
  - Testing and quality assurance
Hooker’s General Principles

- 1: The Reason It All Exists
- 2: KISS (Keep It Simple, Stupid!)
- 3: Maintain the Vision
- 4: What You Produce, Others Will Consume
- 5: Be Open to the Future
- 6: Plan Ahead for Reuse
- 7: Think!
Pre-project Survey

- To find out where you are...