CSE 308, Section 2

Software Engineering

Session 1

General Class Issues

- Dr. R. Kelly (contact info on class Web site)
- Hands-on class
- Requirements
  - CSE219 or CSE260
  - CSE320
  - CSE305 or CSE306
  - CSE333 and CSE336 helpful
  - U4 standing

This is your capstone course
Session 1 - Introduction

Texts

- No official text
- Assigned Web readings
  - Optional text – Head First Object-Oriented Analysis & Design
  - Supplemental text – Head First Design Patterns
  - Supplemental text – UML Distilled

Other reading will be introduced in class

CSE308 Web Site

- www.cs.stonybrook.edu/~cse308/

Course Sections
There are 2 sections of CSE 308 in the Fall 2019 semester. For details for each of the sections, follow the links below:
- Section 1: Tues. 10AM-3:30PM, Storrs M113
- Section 2: Tues. 1PM-3:40PM, Storrs M113

Content
CSE 308 introduces the basic concepts and analysis tools and techniques of Software Engineering. It emphasizes the development of reliable and maintainable software via system requirements and specifications, software design methodologies including object-oriented design, implementation, integration and testing, software project management, utility in documentation, software maintenance, and considerations of human factors.

The course covers the core elements of software engineering and the associated tools and techniques. The course is designed to provide an understanding of the principles and practices of software engineering.

For CSE majors, completion of CSE 210 (or CSE 211G), CSE 220, CSE 281, or CSE 295 is required to enroll in this course. You will also find that CSE 300 and CSE 308, although not required, provide some useful background to CSE 308. Depending on the number of students that have completed the course, you may need to spend more time at some of the course objectives as you understand the course better and work towards the final project.

Objectives
The course will cover the following topics:
- The principles of software engineering
- The role of software engineering in the software development process
- The role of software engineering in the design and development of software systems
- The role of software engineering in the testing and evaluation of software systems
- The role of software engineering in the maintenance and evolution of software systems

The course will cover the following objectives:
1. An ability to perform project planning, requirements analysis, and context testing
2. An ability to write and comprehend software requirements and specifications
3. An ability to design, develop, and maintain software systems
4. An ability to test and debug software systems
5. An ability to evaluate software systems

The course will also satisfy the following course requirements:
- A2.1 Design, develop, test and maintain software systems

© Robert Kelly, 2005-2019
Class Web Site

- [www3.cs.stonybrook.edu/~cse308/index-Section2.html](http://www3.cs.stonybrook.edu/~cse308/index-Section2.html)

- Check it regularly for:
  - Syllabus
  - Office hours / location /
    e-mail addresses
  - Assignments and lecture code
  - Class notes (pdf)
    - Print notes before each class
  - References
  - Lots more

*We do not use Blackboard for this class*

Course Outcomes

- An ability to perform project planning, requirements analysis, and
  system/test design.
- An ability to work as a team to produce software systems that meet
  specifications while satisfying an implementation schedule.
- An ability to produce professional quality oral/written presentations
  of system designs, reviews, and project demonstrations.
Goals

- Apply many skills you’ve learned to one project
- Improve your oral communications skills
- Learn to
  - build a system too large for one person
  - decompose a project into a set of smaller builds
  - work as part of a team
  - build maintainable code
  - design first and code later
  - Use design approaches
  - Use design tools
  - Present your work to a group

Development approach consistent with previous CSE courses

Approach

- Emphasis on thinking and designing
  - Not memorization
  - Not trial and error
- Use a development process that resembles industrial work environments
- Learn to adapt to a different programming environment
- Write code with considerations for
  - Team coordination
  - Long term maintenance
Reading Vs. Doing

- Class will cover software engineering practices
- You learn by reading, listening, discussing, and doing (most important)
- Usually we will discuss a topic in class before you include it in your project – but not always
- We will model most software engineering activities in the classroom
  - Design reviews
  - Code reviews
  - Project Presentations

You will get experience in group software activities

Grading

- A, B, C ... grades
- Grade basis
  - Project
  - Exams
  - Oral communications (class discussions and presentations)
  - Written communications
- One mid-term exam and a brief final exam
- In-progress grades will be available on the class Web site, make sure that you check it regularly

A large class size might make for limited class presentations
Grading Curve

Grading curve (A-F) could be a bell curve or (more likely) bi-modal, depending on project quality.

CSE308 Grading History (Kelly)

<table>
<thead>
<tr>
<th>Grades</th>
<th>Fall 2018 Section 1</th>
<th>Spring 2018 Section 1</th>
<th>Spring 2018 Section 2</th>
<th>Fall 2017 Section 1</th>
<th>Fall 2017 Section 2</th>
<th>Spring 2017</th>
<th>Spring 2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>52.1%</td>
<td>33.0%</td>
<td>46.5%</td>
<td>39.1%</td>
<td>47.9%</td>
<td>54.2%</td>
<td>27.3%</td>
</tr>
<tr>
<td>B</td>
<td>28.2%</td>
<td>32.9%</td>
<td>45.2%</td>
<td>30.3%</td>
<td>40.9%</td>
<td>25.0%</td>
<td>51.5%</td>
</tr>
<tr>
<td>C+, C</td>
<td>14.1%</td>
<td>25.0%</td>
<td>12.2%</td>
<td>13.0%</td>
<td>11.2%</td>
<td>16.7%</td>
<td>0%</td>
</tr>
<tr>
<td>C- and below</td>
<td>5.6%</td>
<td>4.1%</td>
<td>1.2%</td>
<td>17.4%</td>
<td>(all W)</td>
<td>0%</td>
<td>4.2%</td>
</tr>
</tbody>
</table>

- No limitations of a bell curve (no limits on As or failing grades)
- Demonstration of good CS skills needed to pass
- Class interaction helps teams help each other
- No possibility to "hide" within a team
Grading

- Final grades are calculated based on a formula (no subjective grading)
- Formula weights all the components of the class
  - Project -50% (all assignments)
  - Mid-term exam - 25%
  - Final exam - 10%
  - Oral communications - 15%
- Final grade is based on your total score (the higher the score, the higher the grade)

Grade Sheet

- Unofficial class roster
- Check it regularly
- Ask for more feedback if you are not getting enough in class or in e-mail

<table>
<thead>
<tr>
<th>308 ID</th>
<th>Mid-term</th>
<th>Oral</th>
<th>Project</th>
<th>Final</th>
<th>Final</th>
<th>Rank</th>
<th>Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>65.5</td>
<td>87.5</td>
<td>101.7</td>
<td>66</td>
<td>85.9</td>
<td>3</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>67</td>
<td>100.0</td>
<td>91.6</td>
<td>64</td>
<td>83.9</td>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>82.5</td>
<td>100.0</td>
<td>91.6</td>
<td>83</td>
<td>89.7</td>
<td>2</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>71.5</td>
<td>82.5</td>
<td>84.7</td>
<td>83</td>
<td>77.9</td>
<td>17</td>
<td>B-</td>
</tr>
<tr>
<td>5</td>
<td>46</td>
<td>75.0</td>
<td>73.4</td>
<td>57</td>
<td>65.1</td>
<td>21</td>
<td>C</td>
</tr>
<tr>
<td>6</td>
<td>59.8</td>
<td>100.0</td>
<td>90.7</td>
<td>81</td>
<td>82.6</td>
<td>10</td>
<td>A-</td>
</tr>
<tr>
<td>7</td>
<td>37</td>
<td>100.0</td>
<td>91.6</td>
<td>62</td>
<td>77.7</td>
<td>13</td>
<td>A</td>
</tr>
</tbody>
</table>

Grade sheet also includes a ranking column

You will get your CSE308 ID in an e-mail
Session 1 - Introduction

Project Grade Sheet

- Your project is graded incrementally, check it regularly
- Ask for more feedback if you are not getting enough in class or in e-mail

<table>
<thead>
<tr>
<th></th>
<th>Final</th>
<th>Design</th>
<th>Code Review</th>
<th>Final Demo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Req.</td>
<td>Oral Komm.</td>
<td>Quality</td>
<td>Score</td>
</tr>
<tr>
<td>S</td>
<td>NR</td>
<td>72</td>
<td>15</td>
<td>70</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>95</td>
<td>15</td>
<td>88</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>97</td>
<td>15</td>
<td>95</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>97</td>
<td>15</td>
<td>95</td>
</tr>
<tr>
<td>U</td>
<td>S</td>
<td>87</td>
<td>15</td>
<td>85</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>71</td>
<td>15</td>
<td>75</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>65</td>
<td>24</td>
<td>88</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>93</td>
<td>15</td>
<td>85</td>
</tr>
</tbody>
</table>

- Some use cases are required
- Expected use cases might vary with the size of the team
- Final score is the weighted average of the requirements, design review, code review, and final demo

Academic Integrity

- Representing another person’s work as your own is always wrong
  - Assignments
  - Exams
- Gaining an unfair advantage in grading harms other students
- Suspected instances of academic dishonesty will be reported to the Academic Judiciary
- For details, refer to the Academic Judiciary Web site (link on class home page)
Project Team

- Target - 4 total students per team
- Typical team composition
  - Project manager
  - Lead programmer
  - GUI designer
  - Data designer
- Individual responsibilities determined by team
- You will form your own team by 2nd week
  - Be very careful in selecting teammates - your grade might depend on it
- Number of team members and responsibilities of members might vary
- Send me an e-mail with the names of your team members

Project Grading

- Project team grade can vary among team members based on participation in reviews
- Total project grade determined from individual component grades (e.g., code review)
- Points deducted for late submission of components
- Project score will be calculated from
  - design
  - code review, and
  - final project presentation
  - Requirements / test plan
- Think of presentations as oral exams
Project Team Revisions

- Any time during the semester, a project team can request a split of some team members
- If this occurs
  - Each member of the team will have access to the work of the team as of the date of the split
  - Resulting members can continue with a smaller group (with scope revisions) or join another group
- Helps to encourage all team members to work equally hard on project items

Do not wait too long to recommend a revision of your project team

Assignments

- Project submit process will be defined during the semester (once the TAs are assigned)
- Submit once for the group
- Feedback
  - You may get feedback from TAs (time permitting), but remember that your primary project grades are determined during your formal project reviews
  - Good to volunteer in class for mock reviews to get feedback
  - Feedback to other teams mock reviews will be important to your project development
  - Submitting on-time counts to your grade
- Details once TAs are assigned

© Robert Kelly, 2005-2019
Session 1 - Introduction

Oral Communications Grade

- Oral communications
  - will be a factor in your project presentation grades
  - Very important to “volunteer” for in-class presentations
- Components
  - Interim presentations
  - Interaction with other student presentations (e.g., questions)
  - Formal project-related presentations
- Good communications takes lots of practice - the class is the place to get that experience

You will get feedback on your presentation technique as well as your technical content

Being silent is the big risk to your grade

Interim Presentations

- Many steps in the project will involve “volunteer” and required presentations from groups
- Some required presentations will be outside normal class hours
- Presentations model SW group activities
- Presentations allow feedback to improve the project
- Presentations should show preparation, be succinct, and be targeted to the level of the students in the class
- Non-presenters will learn problems to avoid along with solution techniques
Example - Oral Communications

- Analysis of another student's work should:
  - Be offered in a way that is positive and respectful of the other student
  - Show an understanding of the material the other student presented
  - Contribute to the overall class understanding of the material
  - Usually offer a differing opinion of an aspect of the system design or a consideration not previously addressed

Teamwork

- The project will give you good teamwork experience
- Important that all team members contribute to the team efforts
- All team members are expected to contribute during the project reviews
Project development will include:
- DB
- DB Persistence layer
- Reports
- Web GUI
- System logic
- Data generation
- Multi-server coordination
- Testing

Testing and evaluation of results will be an important part of the project.

Spring 2019 Project

- Election District Graph Generation
  - OO Software development
  - Significant requirements analysis
  - Significant algorithmic analysis
  - Supercomputer prototyping

Consider changing sections during the first week if you prefer the Section 1 class.

There will be a 3-stage published requirement specification.
**Project Approach**

- When you develop the project, you will follow standard software engineering practices found in industry today
- Quality software philosophy
  - Outcome measurement
  - Process improvement
  - System improvement
  - Emphasis on iteration based models
- Grading philosophy
  - supports incremental improvement

**Hardware Requirements**

- Software for the course should be available to you for your own computer
- Supercomputer access will be provided to you
- Or - if you need to use a University computer for assignments, studies, etc. you can use a CS Lab
- Your server can be hosted as a localhost or some non-SBU cloud server
  - Your Id and password will be automatically generated
  - Your development and demo environment will not assume testing of concurrency, but your system should reflect concurrency issues
Development Environment

- Server side part of your project will be developed in Java (and supporting languages, libraries, frameworks, etc.)
- Other languages (e.g., JavaScript, Python) will be used for specialized components (e.g., GUI, data preprocessing)
- You can use any Java development environment (e.g., Eclipse, IntelliJ, NetBeans) you are comfortable with, but your IDE
  - Should be compatible with Java 8
  - Should support Java EE 8

Approach to Tools and Libraries

- Tools are not a fundamental part of the course
  - (-) Rare to find a tool dominant over time
  - (-) Tools can lessen understanding of the underlying technology
  - (+) Tools can be very helpful in improving the time for development
- You can use tools / libraries when they facilitate development without lessening understanding
How to Get Help

- Don’t get stuck on a problem - ask for help
- TAs
  - Might be limited (difficult to find TAs with the required skills)
  - TAs will be able to help you use the IDE and answer some programming questions
  - Your TA for assignment submission can be found on the unofficial class roster (next to your ID number)
- Send me general e-mail if you are having trouble
- See me during office hours (or by appointment)

Piazza

- Piazza is a Q&A platform designed to get you answers from classmates and instructors
- It serves as a forum to allow you to collaborate and solve common challenges
- Remember that helping classmates does not negatively impact your grade (no curve)
- You can post any doubts you have or errors you may encounter, and we will post our answers on Piazza directly
- You are also encouraged to answer any questions posted by your classmates. This way when an issue is resolved, everyone gets to benefit and learn from the answer.

Critical mass is important to effective use of Piazza
Lectures

- Lecture slides will be available at the class Web site before each lecture
- Print a copy of the slide handout before class and use it to make notes
- Be sure to review the slides before each exam

Topics

- Software development process
- Software lifecycle
- Requirements
- Use cases
- UML
- Patterns
- Database abstraction

Based on other 308 classes, a key topic is UML sequence diagrams

- Modularity
- Testing
- Code conventions
- Code reviews
- Developing for maintenance
- DB development

Emphasis is on hands-on design and development, not lectures

You are expected to know every aspect of the project development
Your First Assignment

- Send me an e-mail
- E.g., (Hi!, name, id#, “I love Stony Brook”)
- Put “CSE308, Section 1 – HW#1” in the subject line of the e-mail message