General Class Issues

- Dr. R. Kelly (contact info on class Web site)
- Hands-on class - Trans lab
- Requirements
  - CSE 219 or CSE 260
  - CSE 305 or CSE 336 strongly recommended
  - CSE 333 helpful
- Text
  - Optional text - Head First Object-Oriented Analysis & Design
  - Supplemental text - UML Distilled

This is mainly a project course

Other reading will be introduced in class
Class Web Site

- www.cs.stonybrook.edu/~cse308/Section02/
- 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

Goals

- Understand software engineering
- Learn to build a system too large for one person
- Learn to decompose a project into a set of smaller builds
- Apply many skills you’ve learned to one project
- Learn how to work as part of a team
- Learn to design first and code later
  - Design approaches
  - Design tools
- Improve your oral communications skills
**Approach**

- Emphasis on thinking and designing
  - Not memorization
  - Not trial and error
- Use a development process that resembles industrial work environments
- 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
- 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

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>Student ID</th>
<th>Mid-term Exam</th>
<th>Oral Comm.</th>
<th>Project Total</th>
<th>Final Exam</th>
<th>Final Avg.</th>
<th>Rank</th>
<th>Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>80</td>
<td>75</td>
<td>65.8</td>
<td>0.0</td>
<td>100</td>
<td>67</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>77</td>
<td>50.0</td>
<td>91.3</td>
<td>100</td>
<td>84.9</td>
<td>8</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td>95</td>
<td>50.0</td>
<td>77.8</td>
<td>100</td>
<td>80.1</td>
<td>14</td>
<td>B</td>
</tr>
<tr>
<td>6</td>
<td>55</td>
<td>60.0</td>
<td>100.0</td>
<td>95.9</td>
<td>84.2</td>
<td>10</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>55</td>
<td>50.0</td>
<td>77.8</td>
<td>100</td>
<td>79.1</td>
<td>16</td>
<td>B</td>
</tr>
<tr>
<td>8</td>
<td>90</td>
<td>20</td>
<td>75.0</td>
<td>100</td>
<td>81.2</td>
<td>12</td>
<td>B</td>
</tr>
</tbody>
</table>

Grade sheet will also include a ranking column.

You will get your CSE308 ID in an e-mail.

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>Student ID</th>
<th>Design In-class Score</th>
<th>Code In-class Score</th>
<th>Code In-class Score</th>
<th>Final Score</th>
<th>Final Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>50</td>
<td>82.5</td>
<td>92</td>
<td>75</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>80</td>
<td>92</td>
<td>92</td>
<td>75</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>30</td>
<td>62.5</td>
<td>94</td>
<td>92</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>82.5</td>
<td>94</td>
<td>94</td>
<td>92</td>
</tr>
</tbody>
</table>

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.
Session 1 - Introduction

**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
  - Published once approved
- I will assign students to groups based on 1) courses completed and 2) your preferences

© Robert Kelly, 2005-2016
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., DB design).
- Points deducted for late submission of components.
- Project score will be calculated from:
  - design (maximum of design review and design document),
  - code review, and
  - final project presentation.

- Equal weighting (design review is very important because of difficulty).

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.
Assignments

- Submit project components to your TA through the class software repository
- Submit once for the group
- Feedback
  - You will get feedback from TAs (time permitting)
  - Submitting on-time counts to your grade
- Details once TAs are assigned

Oral Communications Grade

- Oral communications
  - Will be a factor in your project presentation grades
  - Might be included as extra points in an exam
  - 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

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
- 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

Project - Electronic Library

- The project is re-engineering of an e-book search system
- Most Initial requirements are in the live-brary site,
- Additional requirements (e.g., DB update) will be supplied
**Project Components**

- Web GUI (multiple roles)
- DB
- Persistence layer
- Logic
- Reports
- Database update
- Multi-server coordination

---

**Project Deliverables**

<table>
<thead>
<tr>
<th>Week</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Project description</td>
</tr>
<tr>
<td>3</td>
<td>Issues document &amp; interface flow diagram</td>
</tr>
<tr>
<td>4</td>
<td>Working interface</td>
</tr>
<tr>
<td>5</td>
<td>Project plan</td>
</tr>
<tr>
<td>6</td>
<td>Design document</td>
</tr>
<tr>
<td>8</td>
<td>Stubbed code</td>
</tr>
<tr>
<td>12</td>
<td>Compilable code &amp; test plan</td>
</tr>
<tr>
<td>15</td>
<td>Final documents, demonstration</td>
</tr>
</tbody>
</table>

*Emphasis is on the process and the quality of design, code, etc.*
**Project Approach**

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

**Trans Lab**

- Software for the course should be available to you for your own computer
- Or - if you need to use a University computer for assignments, studies, etc. you can use the Trans Lab
  - Your id and password will be automatically generated
Session 1 - Introduction

TA Sessions

- Scheduled TA sessions in NCS115 (80 minutes)
- Cover major hands-on topics (Goal is to learn to use the tools you will need to complete your project)
- Sample sessions:
  - Hibernate
  - NetBeans development

Java Development Environment

- Your project will be developed in Java (and supporting languages)
- You can use any Java development environment (e.g., Eclipse, NetBeans) you are comfortable with, but your IDE
  - Should be compatible with Java 8
  - Should support servlet and JSP execution (if your system has a Web interface)
- NetBeans 8.1 is recommended
Approach to Tools

- 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
- We will use tools when they facilitate development without lessening understanding
- We will introduce tools during the semester
- In some cases, a tool will be covered during a TA session

How to Get Help

- Don’t get stuck on a Java / IDE problem - ask for help
- TAs
  - 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)
How to Learn the Material

- Think, think, think
- Code, code, code
- Attend class and review the on-line class notes
- Attend TA sessions in CS teaching lab (learn to use NetBeans, XML Spy, nVu, DreamWeaver, etc.)
- Speak up in class

plan on at least 5-10 hours per week in developing software, interacting with your team, writing documents, etc.

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
Session 1 - Introduction

Topics

- Software development process
- Software lifecycle
- Requirements
- Use cases
- UML
- Patterns
- Database abstraction
- Ethics in software engineering
- Modularity
- Testing
- Code conventions
- Code reviews
- DB development
- Web GUI programming

You are expected to know every aspect of the project development.

Your First Assignment

- Become familiar with Live-Brary (or NYPL)
- Be prepared to begin to discuss requirements in the next class
- Send me an e-mail
  - E.g., (Hi!, name id#, “I love Stony Brook”)
  - Put “CSE308 - HW#1” in the subject line of the e-mail message