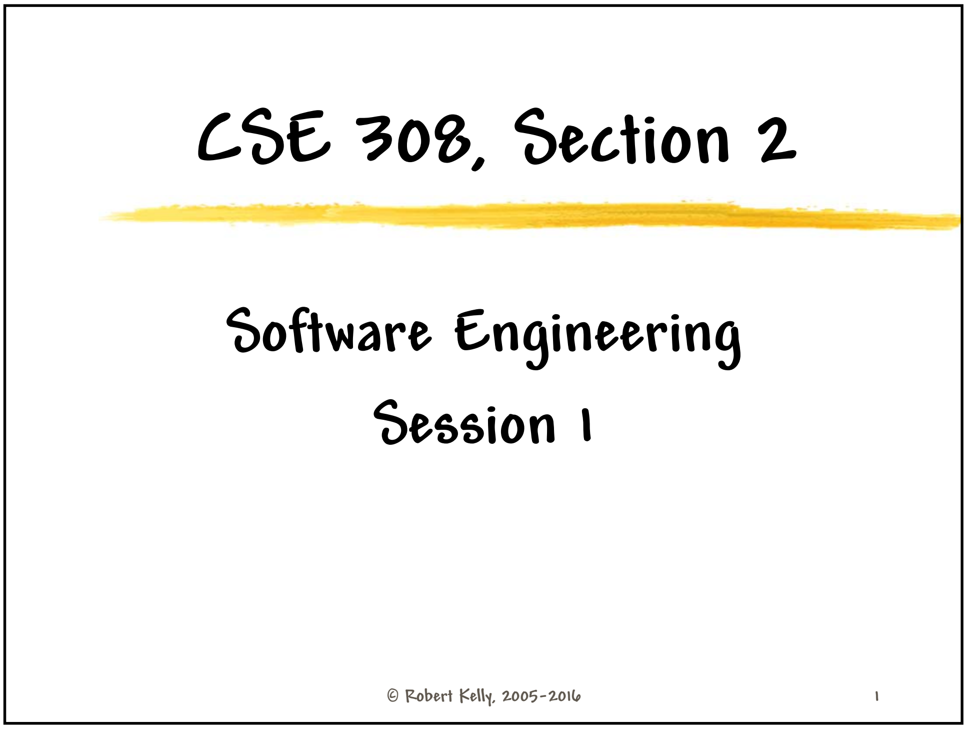


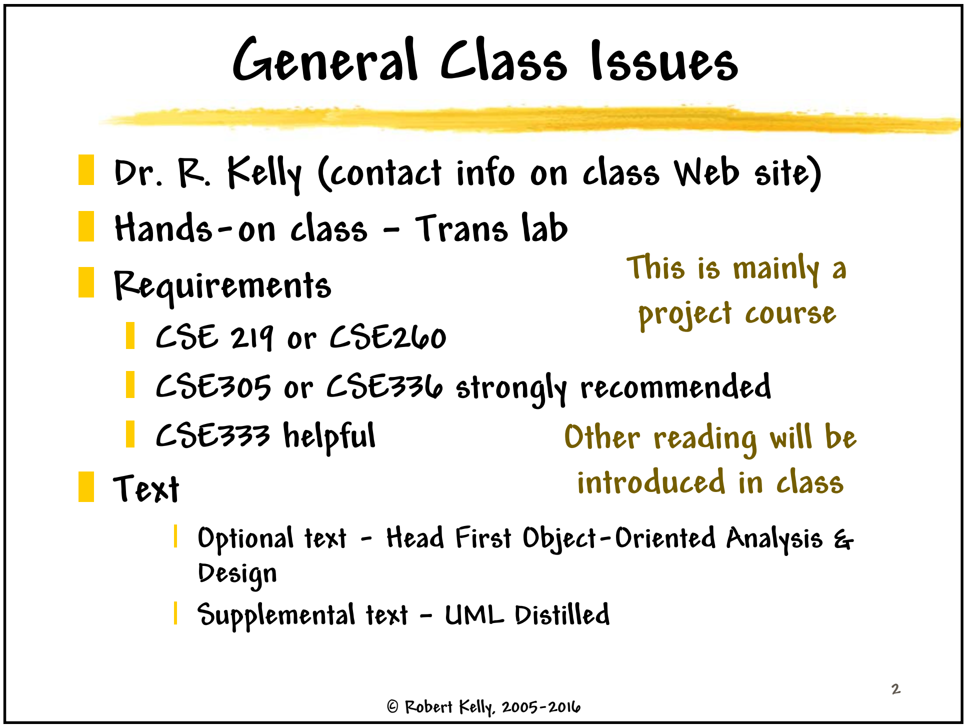
CSE 308, Section 2

Software Engineering

Session 1

© Robert Kelly, 2005-2016

1



General Class Issues

- Dr. R. Kelly (contact info on class Web site)
- Hands-on class - Trans lab
- Requirements
  - CSE 219 or CSE260
  - CSE305 or CSE336 strongly recommended
  - CSE333 helpful
- Text
  - Optional text - Head First Object-Oriented Analysis & Design
  - Supplemental text - UML Distilled

This is mainly a project course

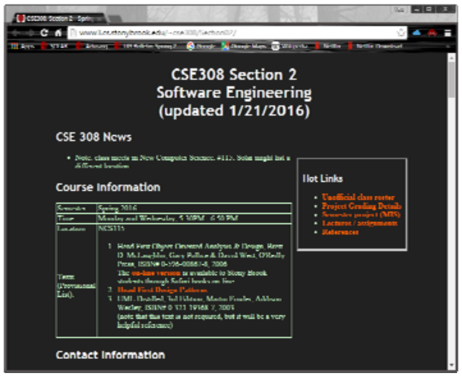
Other reading will be introduced in class

© Robert Kelly, 2005-2016

2

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



CSE308 Section 2 - Spring
www.cs.stonybrook.edu/~cse308/Section02/

CSE308 Section 2
Software Engineering
(updated 1/21/2016)

CSE 308 News

- Note: class meets in New Computer Science, #115. Solar might list a different location

Course Information

Semester	Spring 2016
Time	Monday and Wednesday, 5:30PM - 6:50 PM
Location	NCS115
Texts (Provisional List).	1. Head First Object Oriented Analysis & Design, Brett D. McLaughlin, Gary Pollice & David West, O'Reilly Press, ISBN# 0-596-00867-8, 2006 The on-line version is available to Stony Brook students through Safari books on line 2. Head First Design Patterns 3. UML Distilled, 3rd Edition, Martin Fowler, Addison Wesley, ISBN# 0 321 19368 7, 2003 (note that this text is not required, but it will be a very helpful reference)

Hot Links

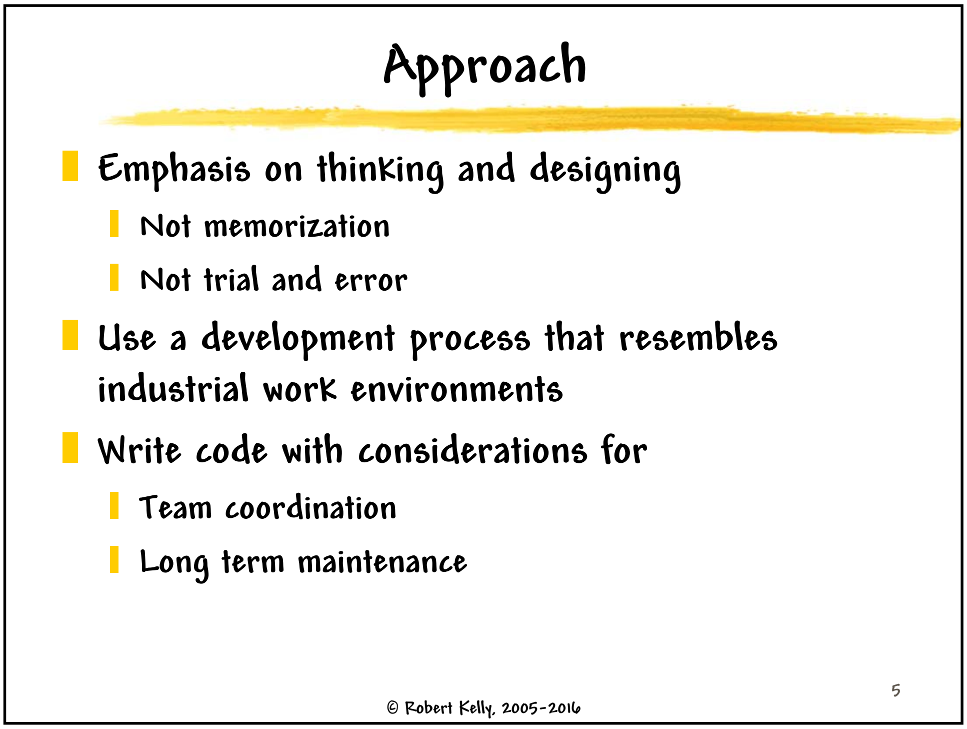
- Unofficial class roster
- Project Grading Details
- Semester project (MBS)
- Lectures / assignments
- References

Contact Information

We do not use
Blackboard for this class

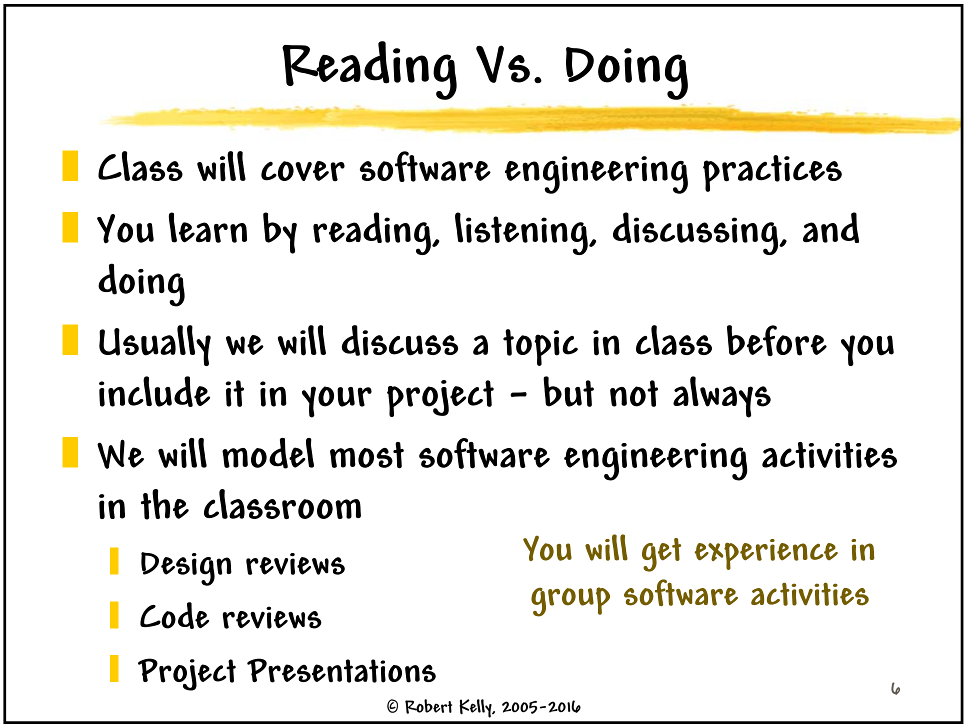
Goals

- 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

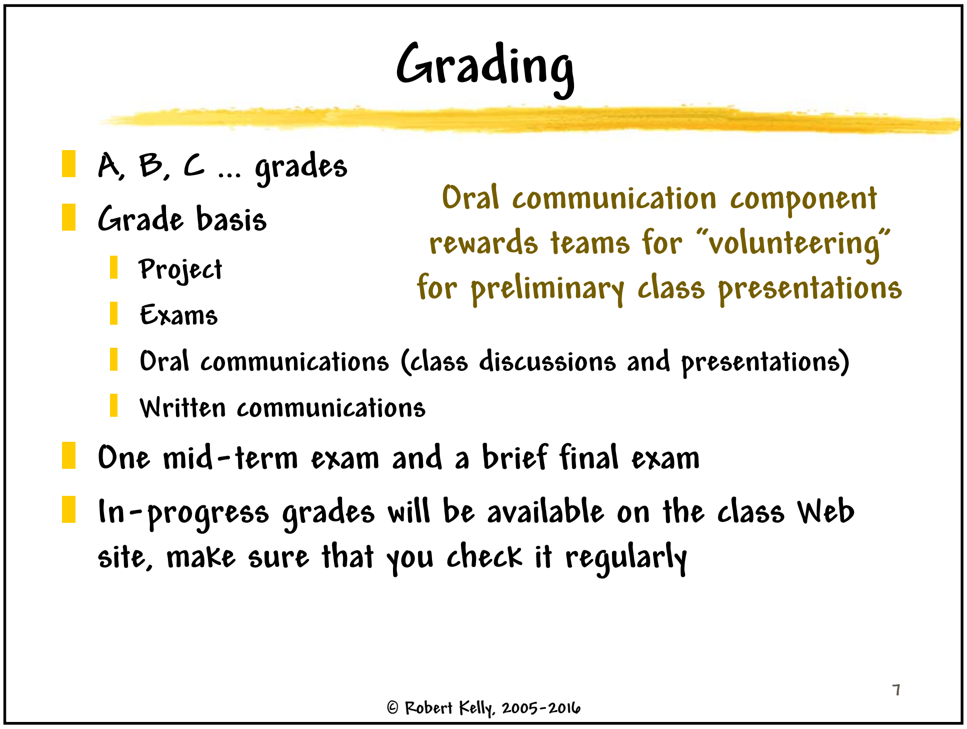
- Emphasis on thinking and designing
  - Not memorization
  - Not trial and error
- Use a development process that resembles industrial work environments
- Write code with considerations for
  - Team coordination
  - Long term maintenance



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



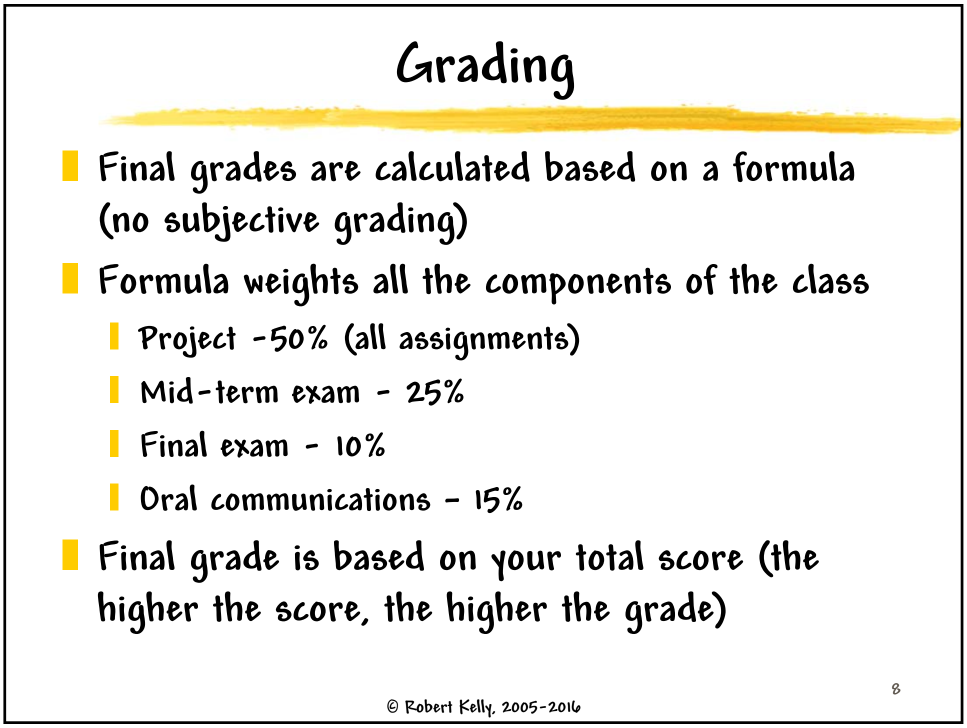
Grading

- 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

Oral communication component rewards teams for "volunteering" for preliminary class presentations

© Robert Kelly, 2005-2016

7



Grading

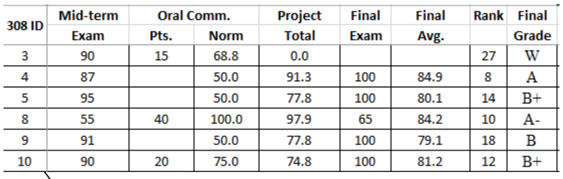
- Final grades are calculated based on a formula (no subjective grading)
- Formula weights all the components of the class
  - Project -50% (all assignments)
  - Mid-term exam - 25%
  - Final exam - 10%
  - Oral communications – 15%
- Final grade is based on your total score (the higher the score, the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

© Robert Kelly, 2005-2016

8

Grade Sheet

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



308 ID	Mid-term	Oral Comm.		Project	Final	Final	Rank	Final
	Exam	Pts.	Norm	Total	Exam	Avg.		Grade
3	90	15	68.8	0.0			27	W
4	87		50.0	91.3	100	84.9	8	A
5	95		50.0	77.8	100	80.1	14	B+
8	55	40	100.0	97.9	65	84.2	10	A-
9	91		50.0	77.8	100	79.1	18	B
10	90	20	75.0	74.8	100	81.2	12	B+

Grade sheet will also include a ranking column

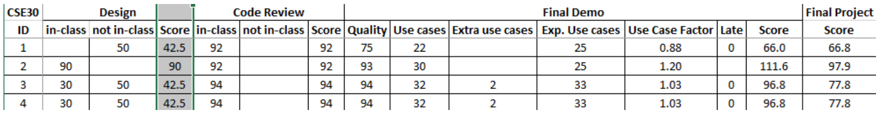
You will get your CSE308 ID in an e-mail

9

© Robert Kelly, 2005-2016

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



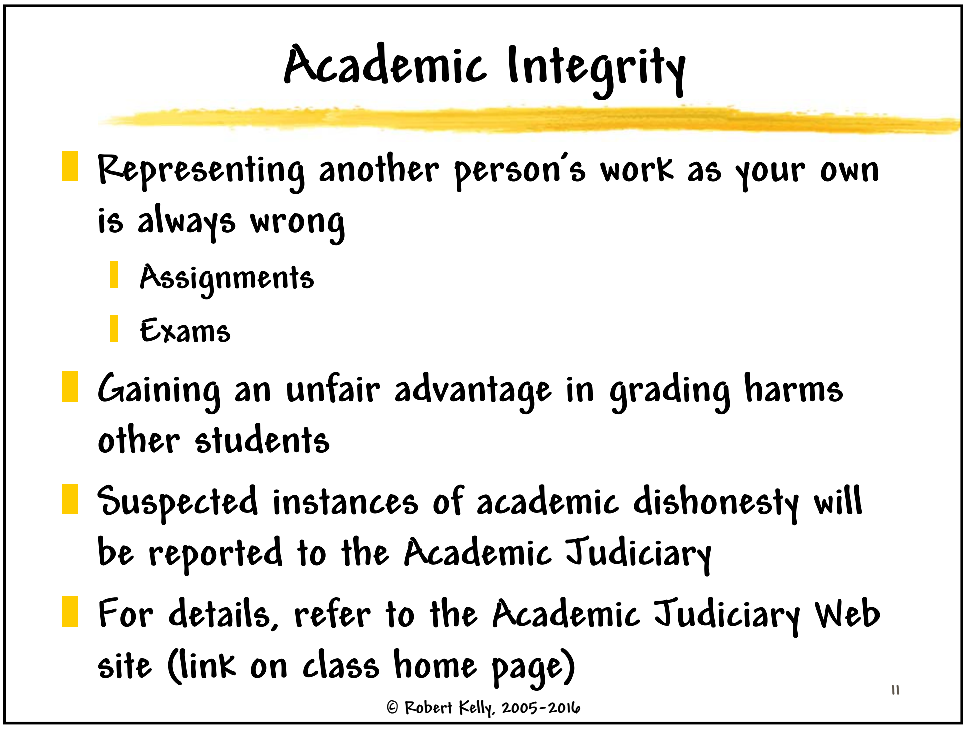
CSE30	Design			Code Review			Final Demo							Final Project
ID	in-class	not in-class	Score	in-class	not in-class	Score	Quality	Use cases	Extra use cases	Exp. Use cases	Use Case Factor	Late	Score	Score
1		50	42.5	92		92	75	22		25	0.88	0	66.0	66.8
2	90		90	92		92	93	30		25	1.20		111.6	97.9
3	30	50	42.5	94		94	94	32	2	33	1.03	0	96.8	77.8
4	30	50	42.5	94		94	94	32	2	33	1.03	0	96.8	77.8

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

10

© Robert Kelly, 2005-2016

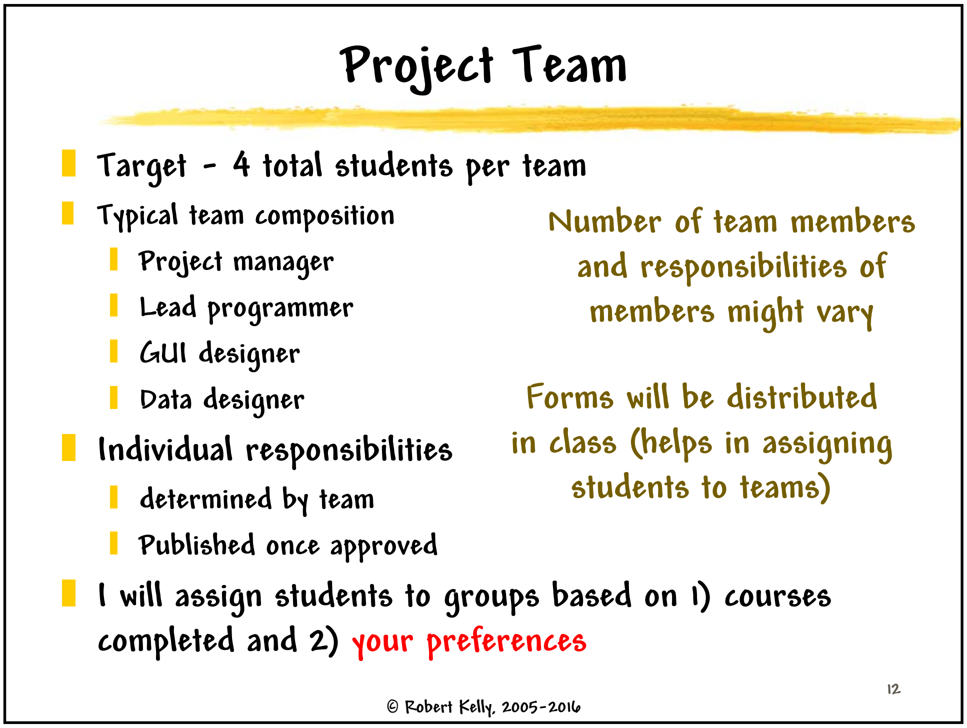


Academic Integrity

- 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)

© Robert Kelly, 2005-2016

11



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

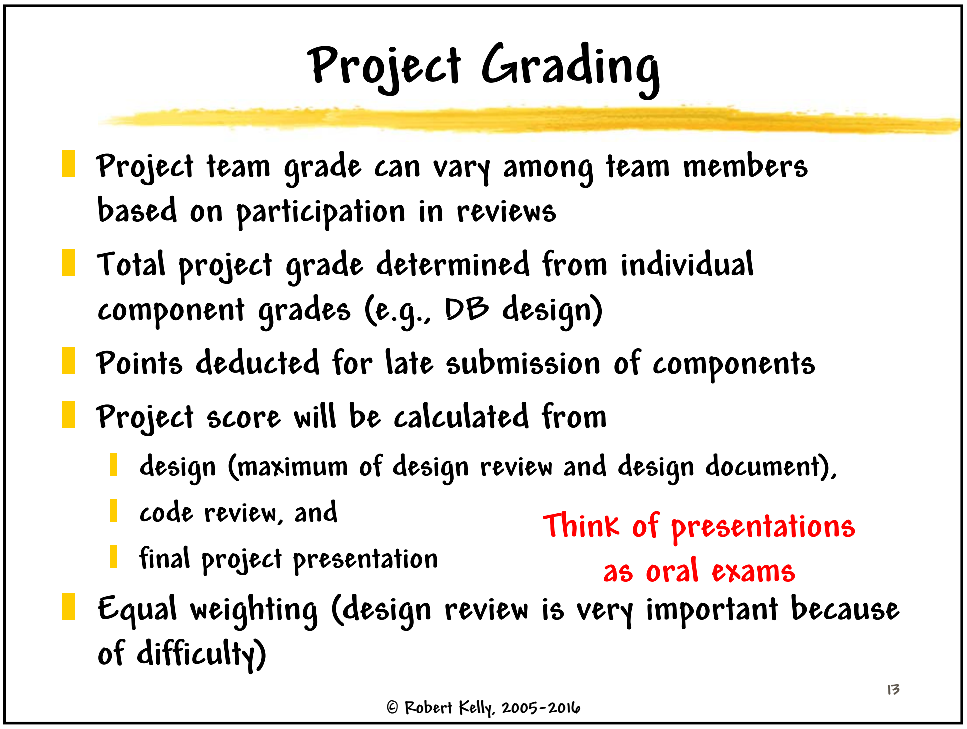
Number of team members and responsibilities of members might vary

Forms will be distributed in class (helps in assigning students to teams)

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

© Robert Kelly, 2005-2016

12

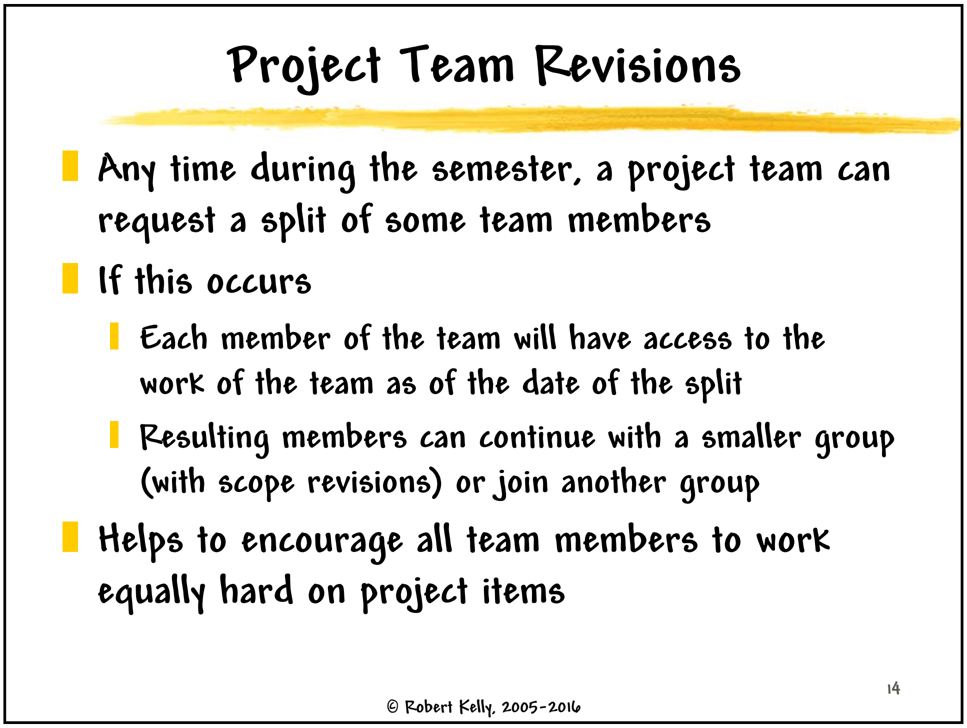


Project Grading

- 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

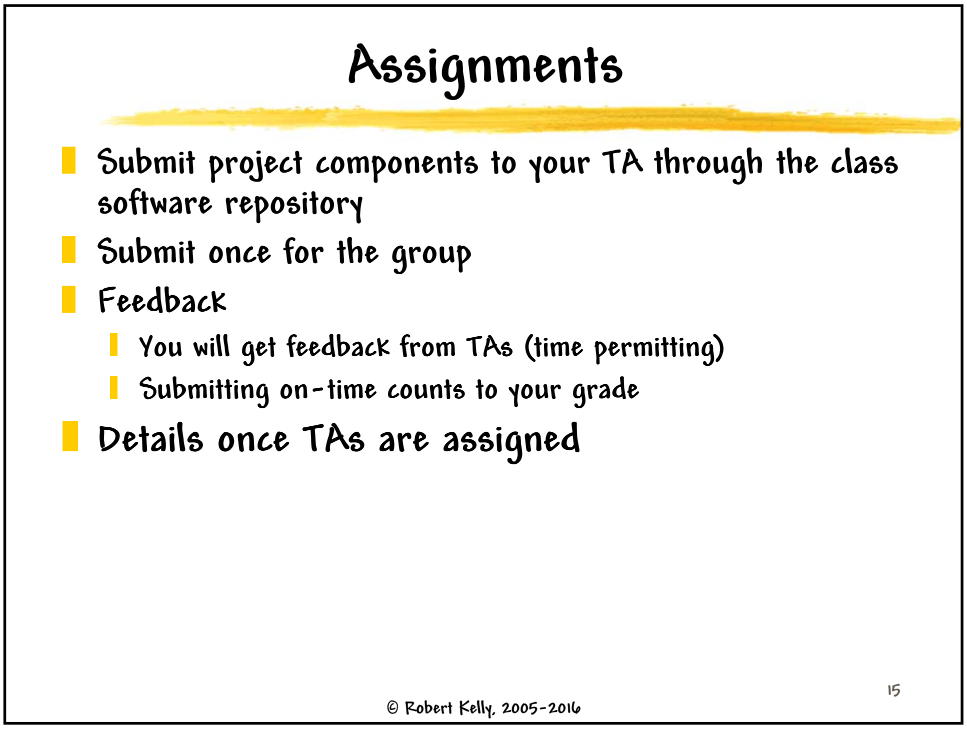
13



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

14

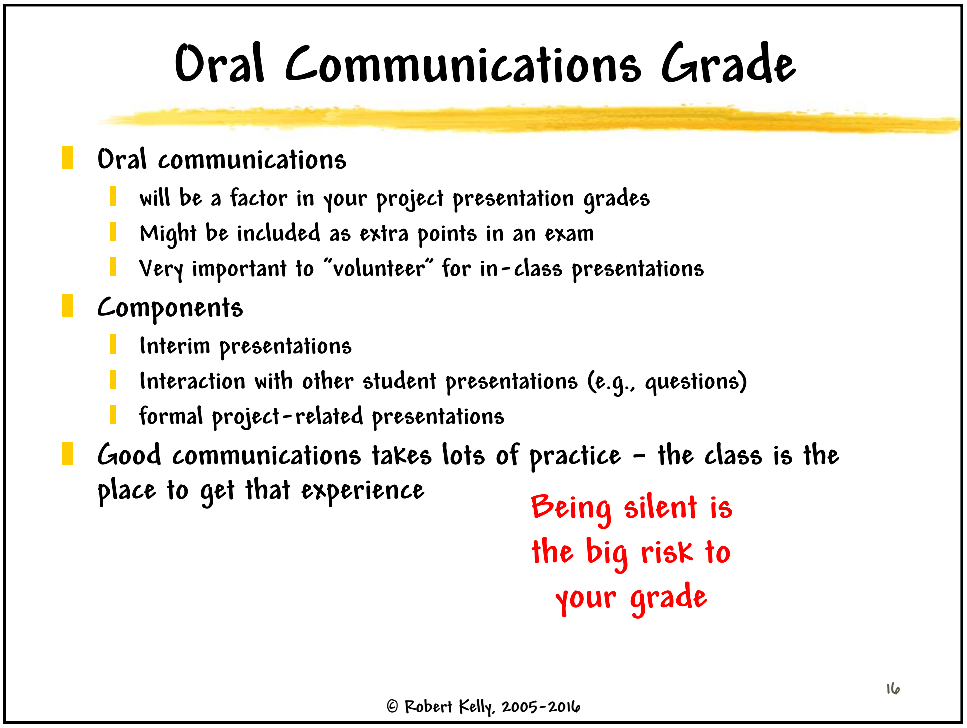


Assignments

- 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

© Robert Kelly, 2005-2016

15



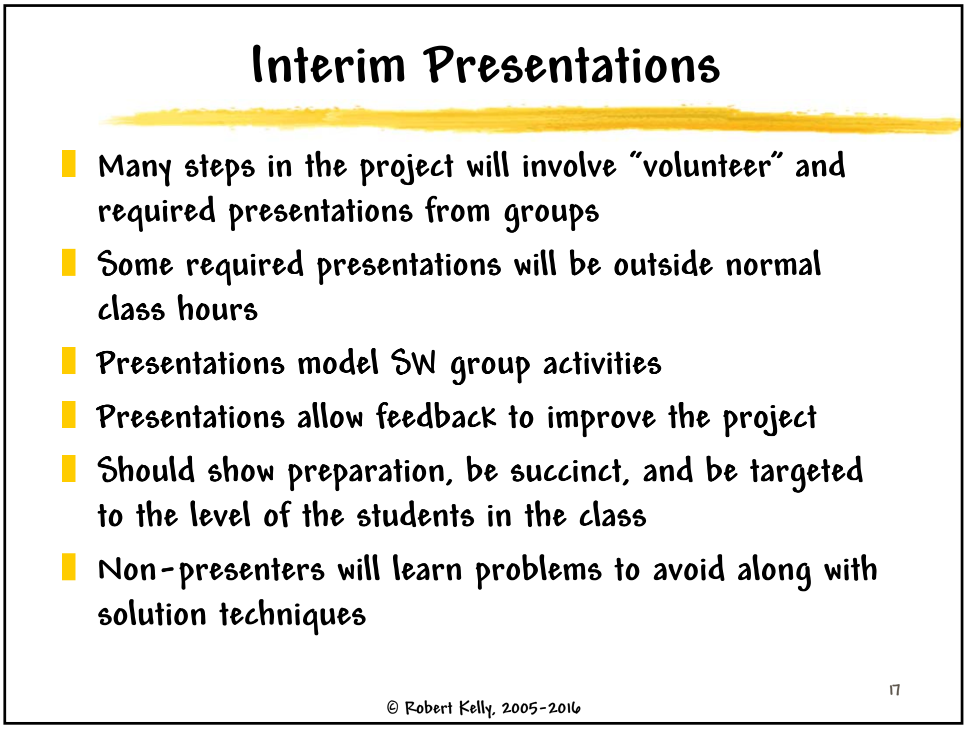
Oral Communications Grade

- 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

© Robert Kelly, 2005-2016

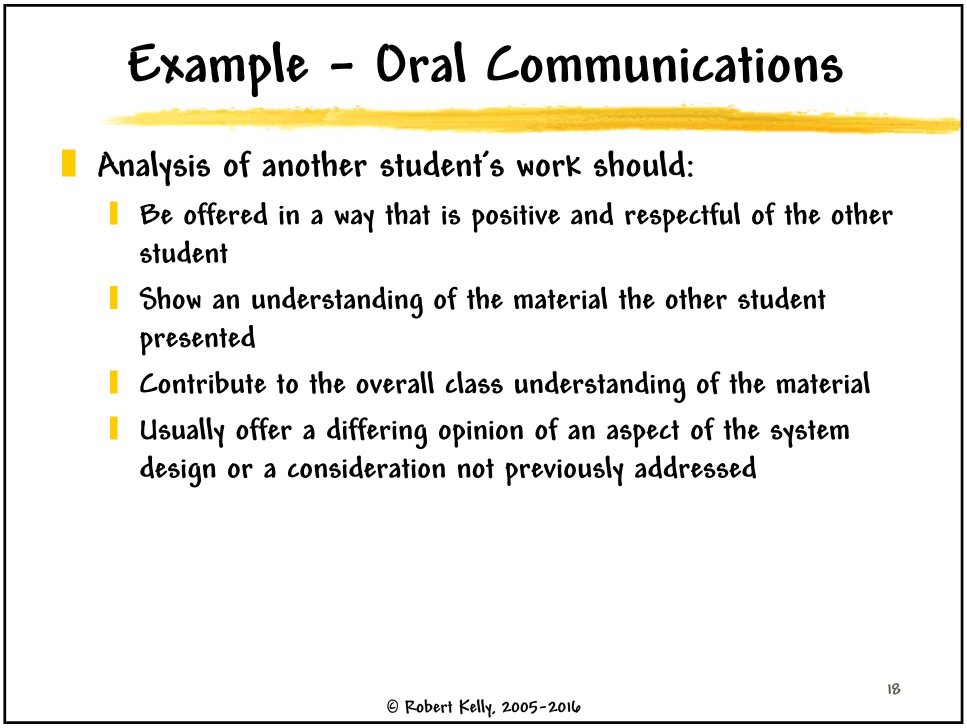
16



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

17



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

18

Teamwork

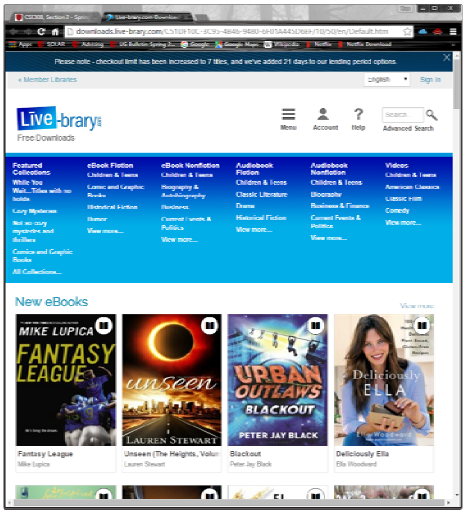
- The project will give you good teamwork experience
- Important that all team members contribute to the team efforts

© Robert Kelly, 2005-2016

19

Project – Electronic Library

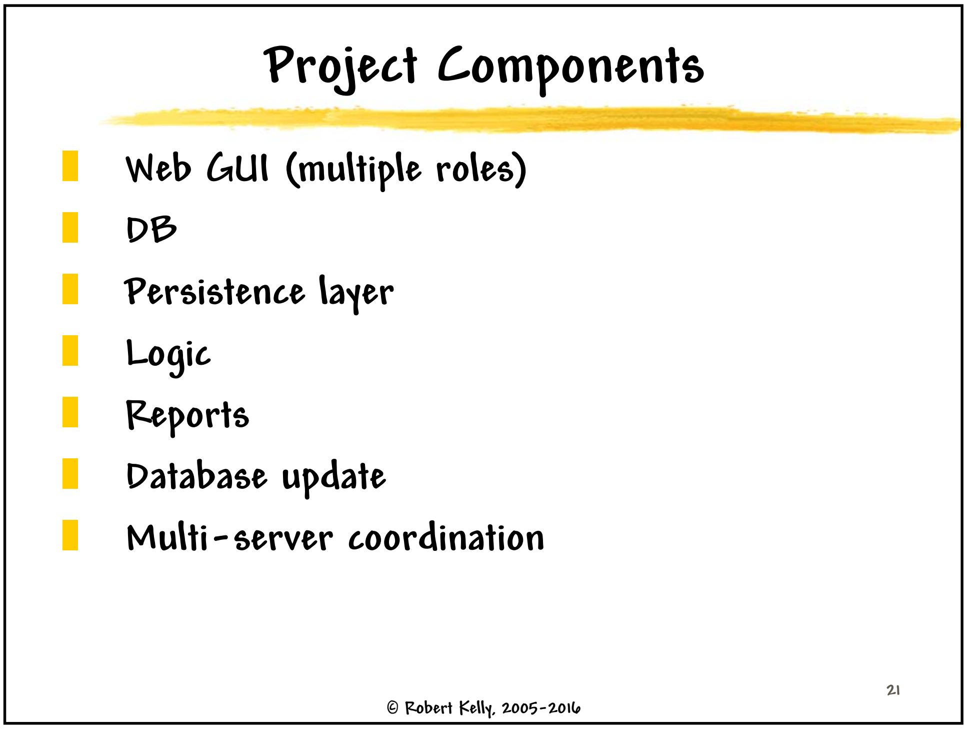
- The project is re-engineering of an e-book search system
- Most Initial requirements are in the live-brary site,
- Additional requirements (e.g., DB update) will be supplied



Please note - checkout limit has been increased to 7 titles, and we've added 21 days to our lending period options.

© Robert Kelly, 2005-2016

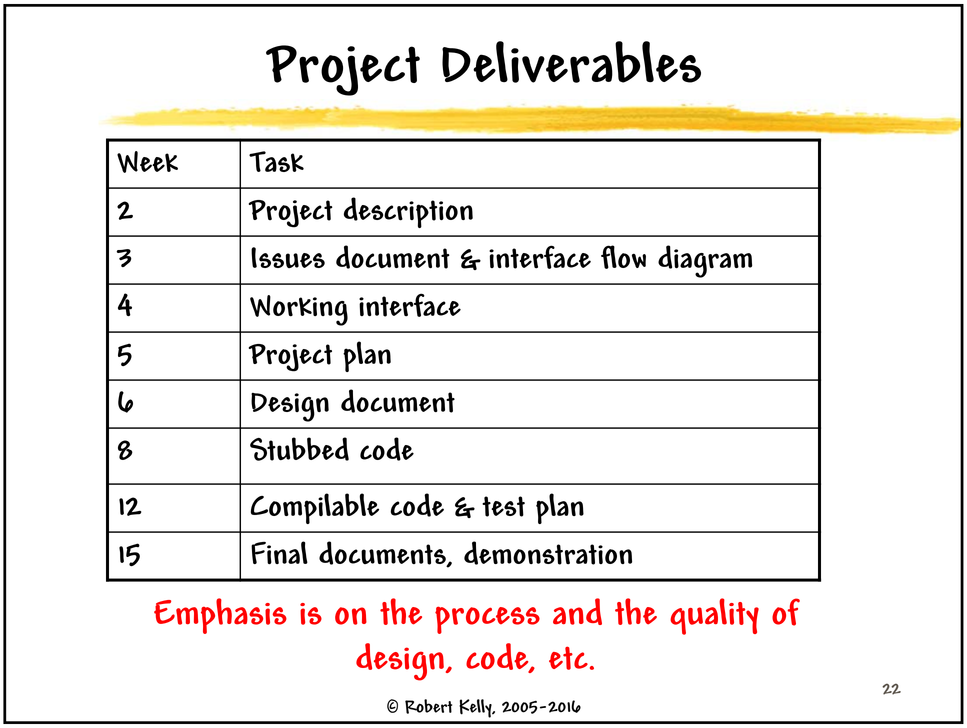
20



Project Components

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

21

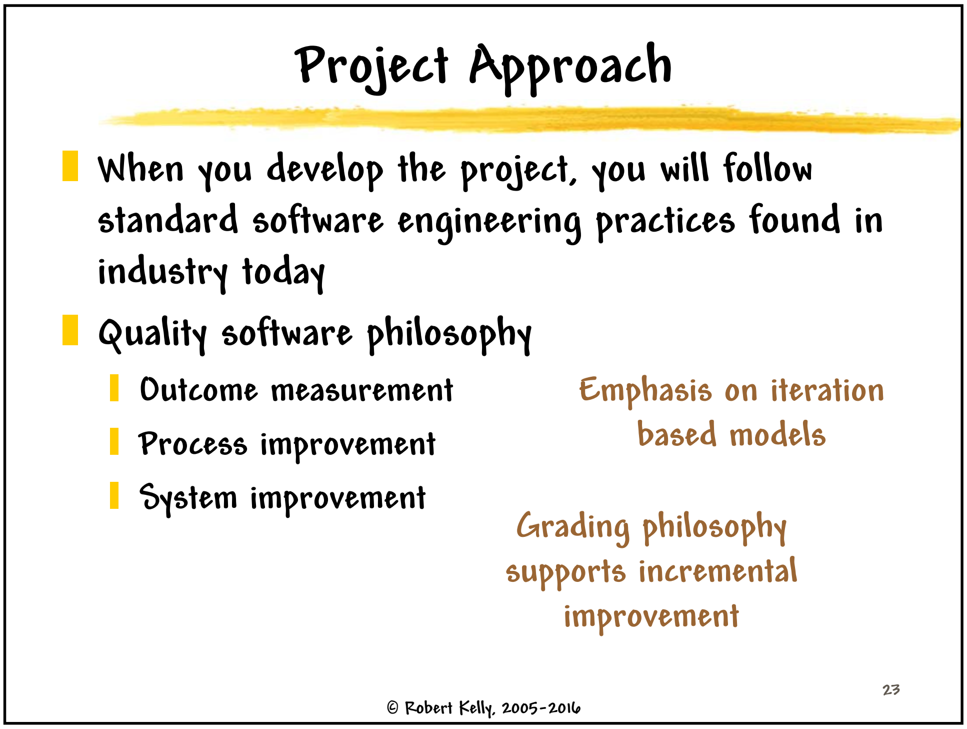


Project Deliverables

Week	Task
2	Project description
3	Issues document & interface flow diagram
4	Working interface
5	Project plan
6	Design document
8	Stubbed code
12	Compilable code & test plan
15	Final documents, demonstration

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

22



Project Approach

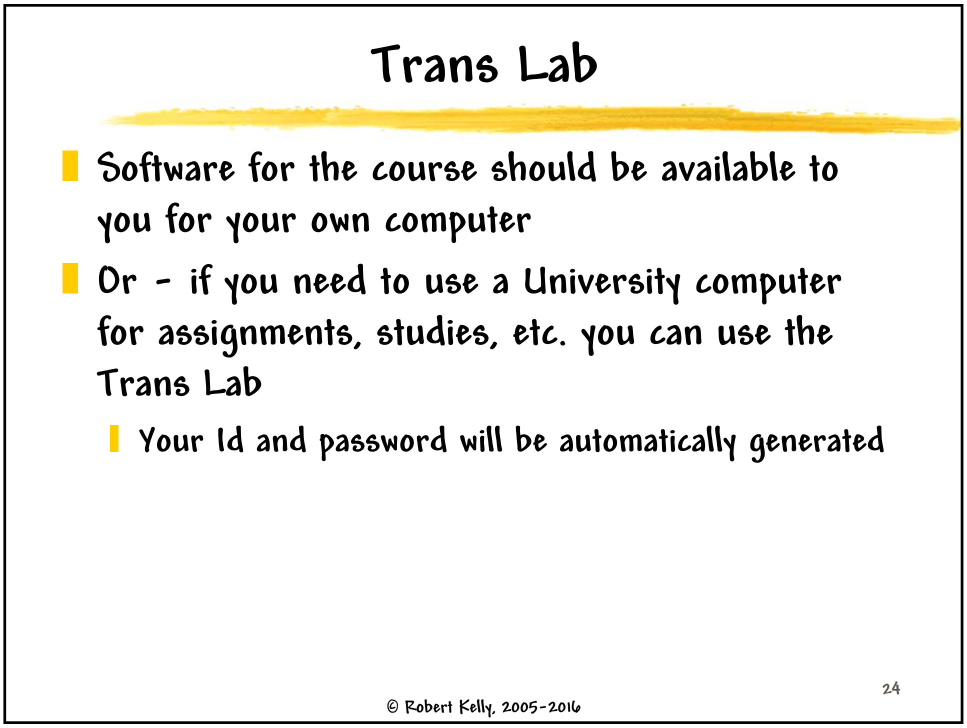
- When you develop the project, you will follow standard software engineering practices found in industry today
- Quality software philosophy
  - Outcome measurement
  - Process improvement
  - System improvement

Emphasis on iteration based models

Grading philosophy supports incremental improvement

© Robert Kelly, 2005-2016

23

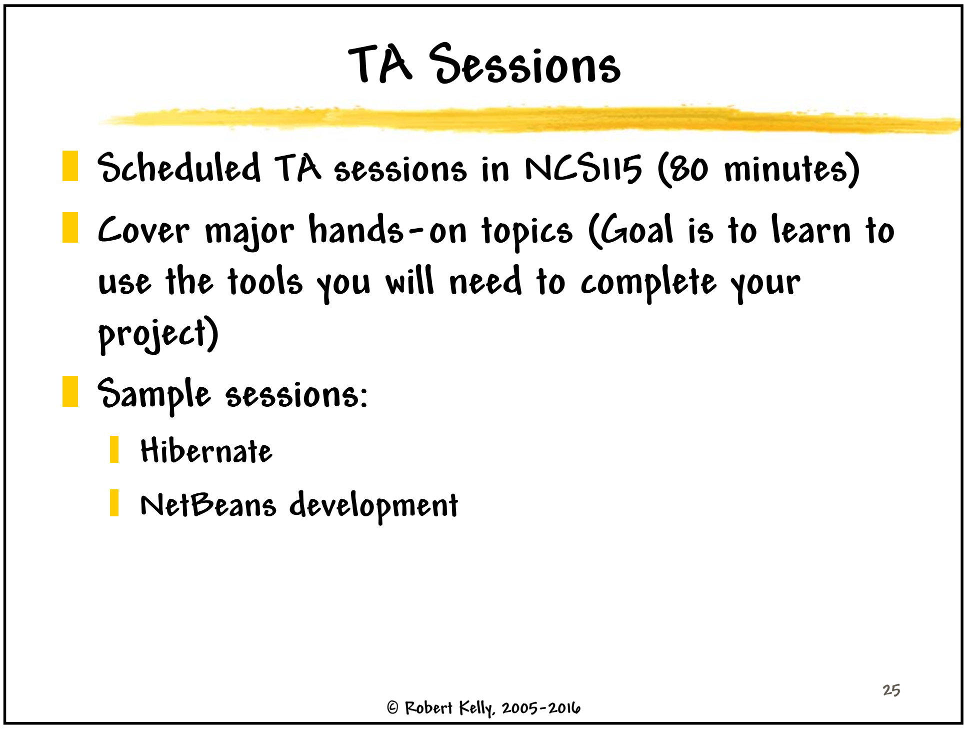


Trans Lab

- 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

© Robert Kelly, 2005-2016

24

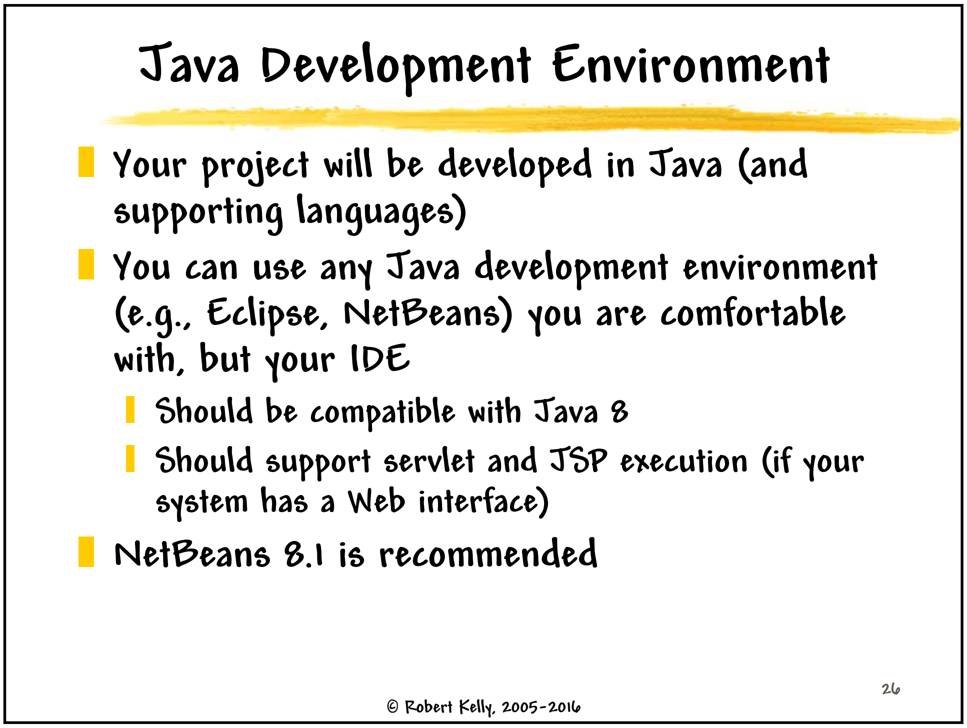


TA Sessions

- 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

© Robert Kelly, 2005-2016

25

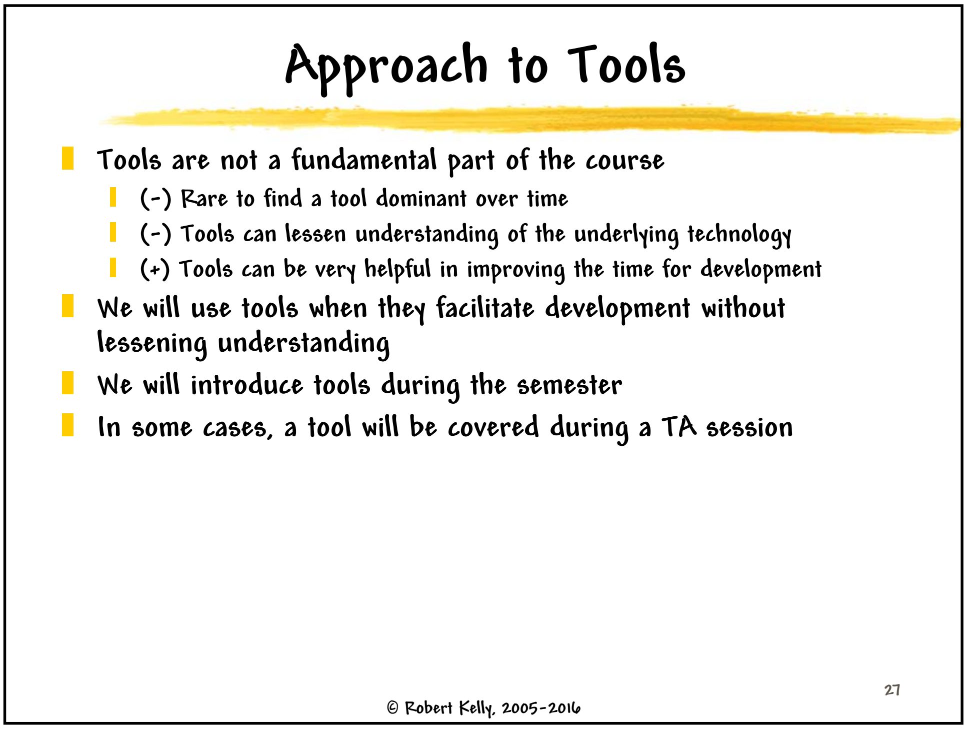


Java Development Environment

- 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

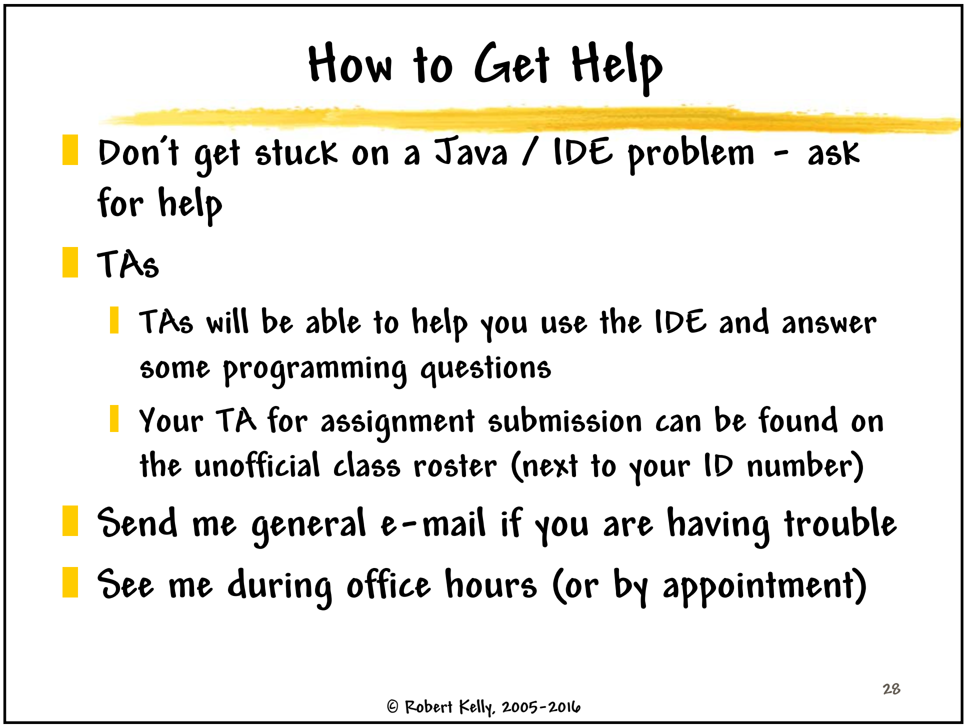
© Robert Kelly, 2005-2016

26



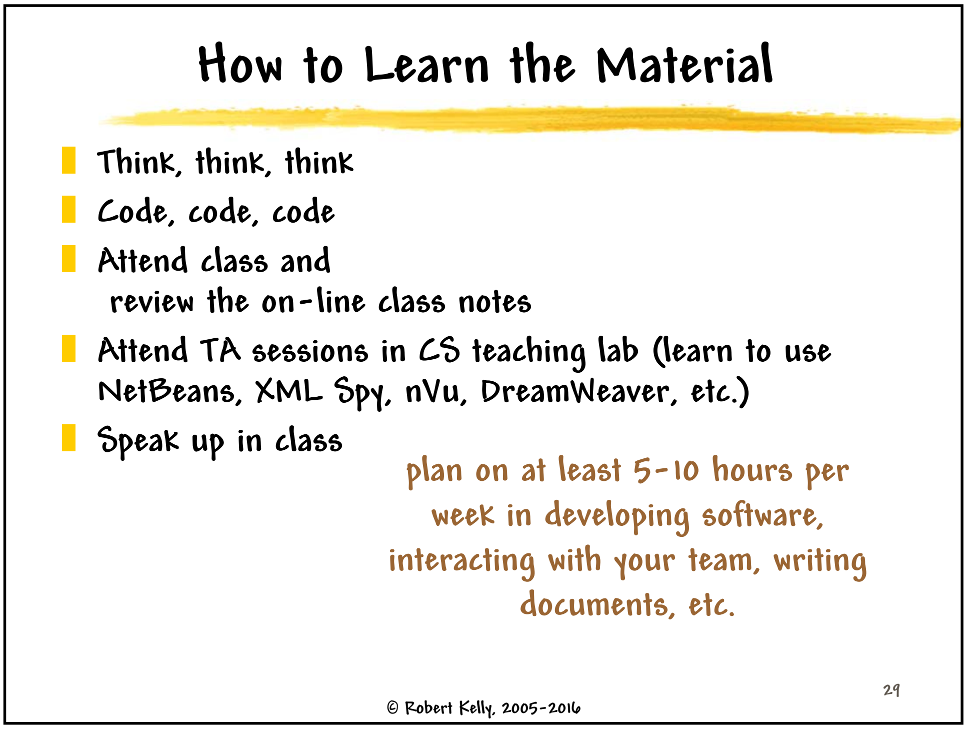
Approach to Tools

- 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)



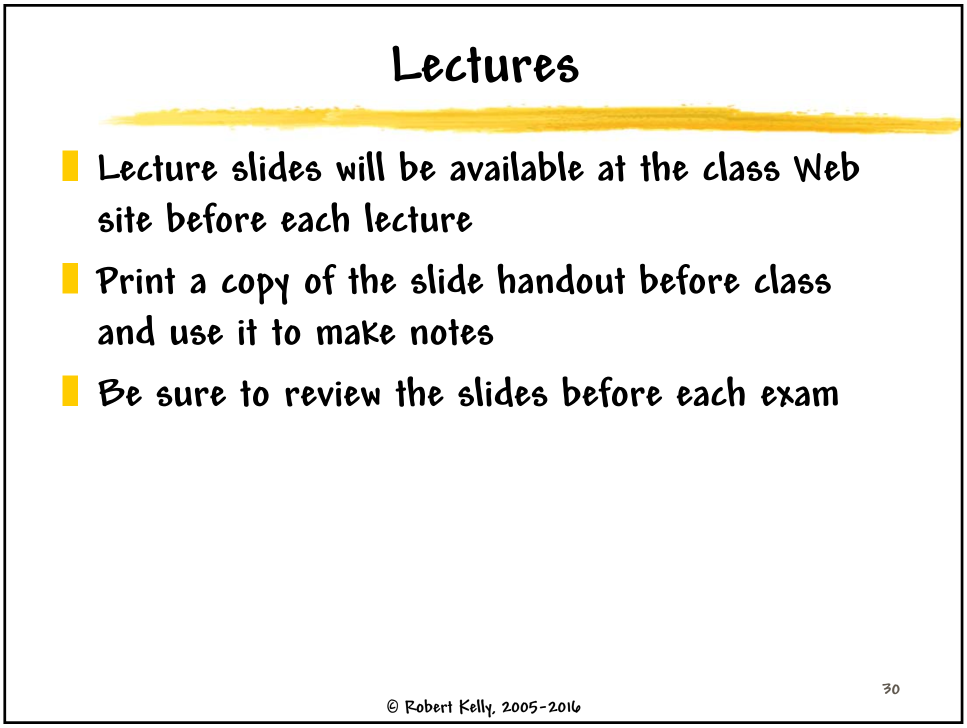
How to Learn the Material

- 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.

© Robert Kelly, 2005-2016

29

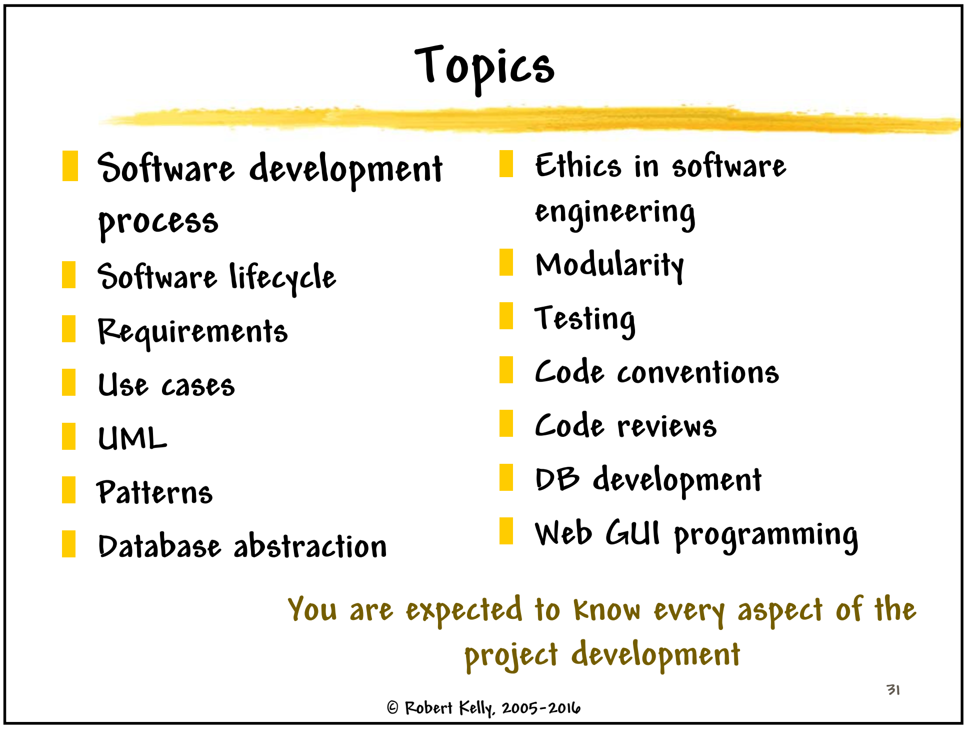


Lectures

- 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

© Robert Kelly, 2005-2016

30



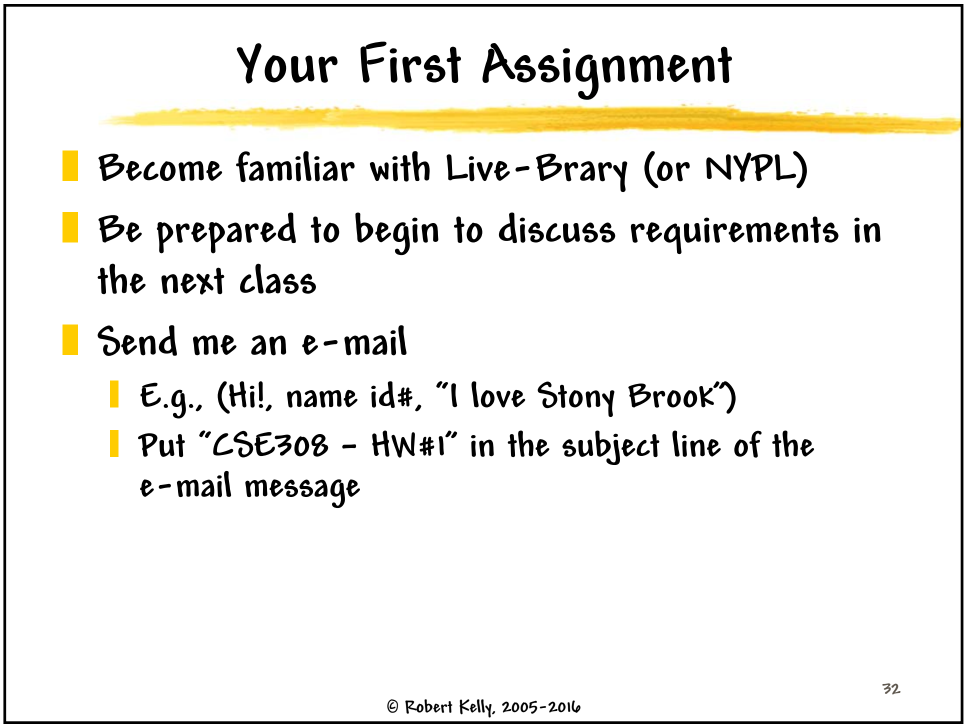
Topics

- 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

© Robert Kelly, 2005-2016

31



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

© Robert Kelly, 2005-2016

32