What’s this course all about?

- [http://inst.eecs.berkeley.edu/~cs61b/su09/](http://inst.eecs.berkeley.edu/~cs61b/su09/)

Project 1

Project 2

Project 3

Staff
Lab Assistants (more to come!)

Stephanie Chou
Jaeon Ki
Dylan Scott

Lectures
Tuesday/Thursday 5:10-6:00
10 Evans

What's the difference between CS61B and CS61BL?

<table>
<thead>
<tr>
<th>1 week in CS61B</th>
<th>1 week in CS61BL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>Lecture</td>
</tr>
<tr>
<td>Lecture</td>
<td>Lab</td>
</tr>
<tr>
<td>Lecture</td>
<td>Discussion</td>
</tr>
<tr>
<td>Discussion</td>
<td>Lab</td>
</tr>
<tr>
<td>Lab</td>
<td>Discussion</td>
</tr>
</tbody>
</table>

Labs
- Online curriculum: ucwise.org
- During lab
  - Take the quiz online (capped at 70%)
  - Do online lab activities
  - Work with a partner
  - Participate in discussion section
  - Get help from other students, lab assistants & TAs

Why are we doing lab based?
- Community
- Learn by doing
- Interviewing practice
- Extra pair of eyes (or ears)
- Frequent feedback/support

- Lecture Notes:
- Lecture Videos:
How is summer session different?

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Lab</th>
<th>Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 week</td>
<td></td>
<td></td>
</tr>
<tr>
<td>In</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS61BL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Do I have to come to lab?

- Yes - to get credit for the quiz!

Books

- *Head First Java*, second edition, by Kathy Sierra and Bert Bates (O’Reilly, 2005);
- *Pragmatic Unit Testing in Java with JUnit*, by Andrew Hunt and David Thomas (The Pragmatic Bookshelf, 2004);

Reading Assignments

Exams

- **Midterm 1**: Tuesday July 7th, 5-6pm in 10 Evans
- **Midterm 2**: Tuesday July 28th, 5-6pm in 10 Evans
- **Final**: Thursday August 13th, 5-8pm in 10 Evans

Review Sessions

- **Review 1**: Sunday July 5th, 1-4pm in 306 Soda
- **Review 2**: Saturday July 25th, 1-4pm in 306 Soda
- **Review 3**: Sunday August 9th, 1-4pm in 306 Soda
Projects

- **Project 1** (individual):
  - due Monday July 13th, 10pm
- **Project 2** (with 0 or 1 partner):
  - due Wednesday July 22nd, 10pm
- **Project 3** (with 1 or 2 partners):
  - due Tuesday August 11th, 10pm

Homework

<table>
<thead>
<tr>
<th>Assigned:</th>
<th>Due:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday during lab</td>
<td>Wednesday before your lab</td>
</tr>
<tr>
<td>Tuesday during lab</td>
<td>Thursday before your lab</td>
</tr>
<tr>
<td>Wednesday during lab</td>
<td>Monday before your lab</td>
</tr>
<tr>
<td>Thursday during lab</td>
<td>Tuesday before your lab</td>
</tr>
</tbody>
</table>

Grading

<table>
<thead>
<tr>
<th>Assignment category</th>
<th># points</th>
<th>percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>All projects</td>
<td>36</td>
<td>18%</td>
</tr>
<tr>
<td>All other homework</td>
<td>24</td>
<td>12%</td>
</tr>
<tr>
<td>All quizzes</td>
<td>20</td>
<td>10%</td>
</tr>
<tr>
<td>Midterm 1</td>
<td>24</td>
<td>12%</td>
</tr>
<tr>
<td>Midterm 2</td>
<td>36</td>
<td>18%</td>
</tr>
<tr>
<td>Final</td>
<td>60</td>
<td>30%</td>
</tr>
</tbody>
</table>

Grades:

- There is no curve!

<table>
<thead>
<tr>
<th>Points</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>190-200</td>
<td>A+</td>
</tr>
<tr>
<td>180-190</td>
<td>A</td>
</tr>
<tr>
<td>170-180</td>
<td>A-</td>
</tr>
<tr>
<td>160-170</td>
<td>B+</td>
</tr>
<tr>
<td>150-160</td>
<td>B</td>
</tr>
<tr>
<td>140-150</td>
<td>B-</td>
</tr>
<tr>
<td>130-140</td>
<td>C+</td>
</tr>
<tr>
<td>120-130</td>
<td>C</td>
</tr>
<tr>
<td>110-120</td>
<td>C-</td>
</tr>
<tr>
<td>100-110</td>
<td>D+</td>
</tr>
<tr>
<td>90-100</td>
<td>D</td>
</tr>
<tr>
<td>80-90</td>
<td>D-</td>
</tr>
<tr>
<td>&lt;80</td>
<td>F</td>
</tr>
</tbody>
</table>

Cheating

- On every homework assignment and project we will be running cheating detection software to compare your solution with the solution of your classmates and with all previous submissions.
- If you’re falling behind at all – please talk to Colleen or your TA!
Cheating versus Collaboration

- **Obvious rules:** You can’t get a solution from the internet by any method. Don’t copy a current or previous student’s solution.

- **No code rule:** You should never have any part of a current or previous student’s homework or project solution in your possession, either electronically or in hardcopy form.
  - **Lab collaboration clarification:** In lab you will write a lot of programs that you will never turn in for homework or a project. For a program that is not turned in, we encourage you to write code with a partner on one computer and then share that collaboratively written code. Always check the day’s homework assignment to make sure you don’t need to turn that code in. If you do need to turn it in, you can talk, but you can not share code.

- **Reusing code:** You may reuse code that you have written. The first time you are writing the code if it is for a homework or project you MUST write it yourself. If not, you may work together with a partner. If you would like to reuse some code that you and your partner wrote in a regular lab that was not part of a homework or project, you may reuse that code. Please make a note whenever this is the case.

- **Looking at code rule:** You can help a classmate with bugs in their homework or projects and in doing so you can look at their code. Don’t copy their code that you see – all work must be your own and all work must be their own.
  - **Don’t show your own:** If you’re helping a classmate find a bug, it is NOT okay to show them your code – whether it is working or not. You should focus only the buggy code in question.

- **Debugging clarification:** You can help someone if they have a logical error in their code. You should:
  - Never touch their keyboard or mouse.
  - Ask them lots of questions! Examples:
    - If they have a run time error:
      - What is happening on the line indicated by the run-time error? What are you trying to do?
      - How could that error happen on that line? What would need to be true? How could those conditions be met?
  - If the program doesn’t do the right thing:
    - What test case(s) demonstrates the error?
    - Can you come up with a simpler test that demonstrates the error?
    - Can you trace through by-hand a simple example that demonstrates the error?
    - Have you run the debugger on the simple test to see when the error is introduced?
    - Have you tested the sub-procedures to make sure that they work?
Cheating versus Collaboration

**Compiling clarification**: If their code doesn’t compile you can help! You should:
- Never touch their keyboard or mouse.
- Try to help them figure it out for themselves. You should just be helping.
- Talk about each syntax error to make sure that it makes sense why the compiler was complaining.

---

Cheating versus Collaboration

**When in doubt — ask rule**: If you are not sure whether a particular interaction is appropriate, talk to Colleen Lewis or your lab TA before you submit the solution.

---

Cheating versus Collaboration

**Give credit rule**: If you receive a significant idea from someone else, clearly acknowledge that student in your solution. Not only is this a good scholarly conduct, it also protects you from accusation of theft of your colleagues’ ideas.

---

Waitlist

- To be enrolled for the course you need to enroll for a section that is not full.
  - This means the 8-11am or 6-9pm labs
  - Do this ASAP!
- If a spot opens up in a the 11-2 or 2-5 sections you can switch into that section.
- Please don’t over-crowd the 11-2 or 2-5 sections!

---

CS Rocks!

**Welcome to CS61BL!**