Transformation of EOSC 211 (Computer Methods in Earth, Ocean and Atmospheric Sciences) or How We Tried A ZILLION New and Wacky Ideas to
a) figure out what to teach,
b) teach it,
c) see if the students learned it, and
d) measure how they felt about the whole process

Rich Pawlowicz and Josh Caulkins
Dept. of Earth and Ocean Sciences,
University of British Columbia

CWSEI End-of-Year Event April 2010
1) What is EOSC211?

- “the MATLAB course” - skills rather than facts
- Structured as
  - 2x1.5 hour lectures, 1x2 hour lab per week (“theory and practice”)
  - Labs (“practice”) require e-submission of code for (semi) automatic run-testing and marking
  - Assignments (“real problems”) require hard-copy submission of code plus figures.
  - Midterm and Final exams.
Demographic info

- all own computers (30% more than 1)

- Use a computer at least once a day (mostly more)

- 66% have never programmed, but 10% “can write large programs”
2) What was wrong with it -
Student view (via focus group last spring)

- Do less (e.g. fewer labs/assignments) but more in-depth.
  There is simply too much stuff.

- Provide a stronger outline for the course, course goals and
  a stronger overview/introduction in the first few weeks of
  the course.

- Students did not know what to study for the exams.

- Change the weekly structure to include more lab/computer
  time (teach in front of computer).

- Somehow make the assignments more relevant (more ES
  stuff).
2) What was wrong with it - Instructor view

- Hard to get to 'real' programs of any substantial length written.
- Wide (but unknown) range of student ability coming into the class.
- Towards the end of term, lab attendance dropped dramatically
- Workload complaints constant, but valid? necessary?
- Can we short circuit the hours spent staring at a screen debugging?
3) What we changed

- Course structure
  - Learning goals document (CurricCom feedback)
  - “teach the goals”
  - Reduced workload – 7 labs, 3 assignments (from 12 labs, 5 assignments)
  - removed math content to concentrate on programming

- Collaborative learning
  - 'Pair programming' in labs (and eventually in assignments)

- Classroom engagement
  - Name sticks
  - Worksheets
4) How we measured it

- Surveys
  - Pre/post test
  - Midterm Evaluation
  - EOS attitude survey

- Workload assessments
  - Self-reported on labs and assignments
  - VISTA submission time stats
  - Inter-year mark comparisons (labs, midterm)

- Lab TA/instructor checklists

- STLF operations
  - Classroom observations
  - Post-class interviews
  - Focus group

- Unsolicited comments
Results - lab marks

![Graph showing average marks for lab 3 to lab 9 for 2008 and 2009]
How long did the labs take?

lab04: mean = 2.45, n=55

lab05: mean = 3.39, n=48

lab06: mean = 2.43, n=54

lab07: mean = 3.25, n=51

lab08: mean = 2.93, n=51

lab09: mean = 2.88, n=13
Did it take them less time?

![Graph showing self-reported time vs. 2009 Mean Submission Time (hours after start of Lab)]

- **Tuesday Lab**
- **Wednesday Lab**
- **Mean**

Self-reported Time

2009 Mean Submission Time (hours after start of Lab)

2009-2008 Submission Times
So...pair programming results in:

- Labs are done about 15% faster
- Lab Marks are about 10% higher
- ...but (and?) students are MUCH happier.
5) Results - Midterm grade

2009 Midterm
Mean = 31.78 ± 5.72
Median = 32.50

2008 Midterm
Mean = 31.41 ± 7.84
Median = 34.00
Where to go next?

- Made assignments 'pair programmable' (if desired), but final is now 'must-pass'.
- Assignments were more complex than in previous years (no concurrent labs)
- Anecdotal impressions – lab marks are 'tighter' – less really bad ones, not so many really good ones
- “the first bad answer” propagates around computer room.
- .....for more info, go to the interviews...
Which of the following items prepared you for the midterm? List in rank order of importance

- Class worksheets
- Lab material
- Class lectures
- Readings

Percentage of Student Responses

- First Choice
- Second Choice
- Third Choice
- Fourth Choice
I find the worksheets provided during lecture useful to my learning.

- **Agree**: 80%
- **Neutral**: 10%
- **Disagree**: 0%
I would recommend pair-programming to other students
### Summarized Results:

**Main Point of Lecture** – They get it.

**Pace of lectures** – Good (maybe a bit slow).

**Clarity of lectures** – Fine, clear.

**Readings** – 30% of students do >50% of readings. 70% of students do <50% of readings.

**Worksheets** – all students like them, find them very useful.

**Pair-programming** – 80% like it, 20% mixed

**Change one thing?**
1. More computer time
2. Make assignments shorter
3. Need clearer instructions on labs/assign.
Student Problem Checklist
- collected during lab
- preliminary results

B – Background Problems

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1 Tools:</td>
<td>Problems with the PC or Mac, OS X / Linux or other OS, directories (lost files), or other basic tools</td>
</tr>
<tr>
<td>B2 Understanding the task:</td>
<td>Problems understanding the lab exercise / task or its “solution”</td>
</tr>
<tr>
<td>B3 Stuck on program design:</td>
<td>Understand the task / solution but can’t turn that understanding into an algorithm, or can’t turn the algorithm into a program</td>
</tr>
<tr>
<td>B4 Hasn’t read the lab:</td>
<td>Student has not read the lab</td>
</tr>
</tbody>
</table>

G – General Problems

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1 Problems with basic structure:</td>
<td>They have a general design and classes but are getting basic structural details wrong</td>
</tr>
<tr>
<td>G3a Problem naming things:</td>
<td>They have problems choosing names for things.</td>
</tr>
<tr>
<td>G3b Problem naming things:</td>
<td>Have not thought through consequences of name choice</td>
</tr>
<tr>
<td>G4a Trivial mechanics:</td>
<td>Trivial problems with little mechanical details</td>
</tr>
<tr>
<td>G4b Trivial mechanics:</td>
<td>Syntax issues (using round vs. square brackets, forgetting the dot, etc.)</td>
</tr>
<tr>
<td>G5a Matlab issues:</td>
<td>Use of help or online documents</td>
</tr>
</tbody>
</table>

S – Specific Problems

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 Control flow:</td>
<td>Problem with basic sequential flow of control, the role of the main or init method.</td>
</tr>
<tr>
<td>S2 Loops:</td>
<td>Conceptual and practical problems relating to repetition, loops</td>
</tr>
<tr>
<td>S3 Selection:</td>
<td>Conceptual/practical problems relating to selection, if else, switch</td>
</tr>
<tr>
<td>S4 Booleans and conditions:</td>
<td>Problems with booleans, truth values, boolean expressions</td>
</tr>
<tr>
<td>S7 Data flow and method header mechanics:</td>
<td>Especially conceptual problems with arguments / parameters and return types / values.</td>
</tr>
<tr>
<td>S8 Terminal or file IO:</td>
<td>Problems with terminal or file IO / data flow</td>
</tr>
<tr>
<td>S9 Strings:</td>
<td>Strings and string functions.</td>
</tr>
<tr>
<td>S10a Arrays:</td>
<td>Problems in distinguishing between values in an array &amp; indices to them.</td>
</tr>
<tr>
<td>S10b Arrays:</td>
<td>Problems with logical indexing to arrays</td>
</tr>
<tr>
<td>S11 Variables:</td>
<td>Problems with the concept of or use of variables.</td>
</tr>
<tr>
<td>S12 Visibility &amp; scope:</td>
<td>Problems with data field visibility, local variable scope, and namespace / imported package problems</td>
</tr>
<tr>
<td>S13 Expressions &amp; calculations:</td>
<td>Problems with arithmetic expressions, calculations, notation such as “++” and all forms of precedence</td>
</tr>
<tr>
<td>S14 Data types &amp; calculations:</td>
<td>Problems caused by failing to understand different data types and casting for primitive types</td>
</tr>
<tr>
<td>S15 Reference types:</td>
<td>Problems arising from a failure to understand the concept or use of reference types, or that reference types behave differently from primitive types</td>
</tr>
</tbody>
</table>

O - Other

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1 - Other:</td>
<td>I'm too busy to figure out which problem the student is having</td>
</tr>
<tr>
<td>O2 - Other:</td>
<td>Problem not on list (write out brief description of problem below)</td>
</tr>
</tbody>
</table>