Teaching programming and modelling skills to first-year earth & environmental science students

Jenny A. Fisher, Chris Brewer, Yoke Berry, Glennys O’Brien, Richard Cook
University of Wollongong Australia

jennyf@uow.edu.au @AtmosJennyF jennyfisher.github.io/computing-modelling-earthsci/
The Problem

Computing and programming skills are necessary for all scientists.

- large datasets
- 2D/3D visualisations
- numerical models
- complex statistics
- and more...

Traditional geoscience curricula do not teach these skills.

Our solution:
Embed these skills in 1st year Earth & Environmental Sciences core
The Starting Point

Surveys of confidence, interest, & (perceived) usefulness (Hoegh & Moskal, 2009).

Positive sentiment: 83%  60%  86%
Implementation

In-class exercises in Jupyter Notebook:

- Weeks 1-4: computing fundamentals
  e.g.:
  - simple calculations
  - making plots
  - using logic (if statements)
  - repeating code (for loops)
  - writing functions
Implementation

In-class exercises in *Jupyter Notebook:*

- Flow In
- Reservoir
- Flow Out

**Weeks 1-4:** computing fundamentals

**Weeks 5-8:** simple box models

1. Bathtub (basic concept)
2. Daisyworld
3. Population
4. Carbon Cycle
Implementation

Weeks 1-4: computing fundamentals  
Weeks 5-8: simple box models  
Weeks 9-12: creative group projects

e.g.:  
How does evolution impact Daisyworld?  
How much would a switch to electric cars decrease CO₂?  
Could a plague save non-renewable resources?  
What happens to food if bees disappear?  
How will different emission scenarios change future CO₂?
Lessons Learned

1. Need to know how they will use skills in future - panel of experts in lecture

2. Need incentives to come prepared - post-lab review quizzes
Implementation

Each exercise followed by Review Quiz for guided, individual practice

Instantaneous feedback + hints point to online resource trinket.io

Share Code from any Device
Trinket lets you run and write code in any browser, on any device. Easily share or embed the code with your changes when you’re done.
1. Need to know how they will use skills in future - panel of experts in lecture

2. Need incentives to come prepared - post-lab review quizzes

3. Need early & frequent positive feedback - embedded peer review
Weeks 1-8: Peer review using Moodle “workshop”

Student assessment view:

### Week 9 Peer Review

**Assessed submission**

*WEEK 9*

Submitted on Thursday, 30 September 2010, 4:02 PM

- [Week9_Exercise.html](#)

**Instructions for assessment**

Provide marks and feedback for your selected submissions.

| Your assessment | by 8090102_2818 Test student 0
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Not assessed yet</td>
<td></td>
</tr>
</tbody>
</table>

**Assessment form**

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Readability The program was well-organized and structured, with appropriate white space. Steps appeared in a logical order, and code was not repeated unnecessarily.</td>
<td>Never</td>
</tr>
<tr>
<td>2 Comments The program provided enough comments for someone familiar with the basics of Python to understand what each line was doing.</td>
<td>Never</td>
</tr>
<tr>
<td>3 Variables The variables had logical names and were accompanied by comments that describe what they represent and (where relevant) what their units are.</td>
<td>Never</td>
</tr>
<tr>
<td>4 Plots The plots use the correct variables for the x (independent) and y (dependent) axes. They have the correct labels on both the x and y axes and an appropriate title. If multiple lines are shown, a legend is included for at least one of the plots.</td>
<td>Never</td>
</tr>
<tr>
<td>5 Problem Solving The program solved the problem(s) posed in the exercise. This includes printing or plotting the required outputs, using the correct tools if specified in the problem, and using the correct inputs.</td>
<td>Never</td>
</tr>
<tr>
<td>6 Interpretation &amp; Justification The writer thought questions at the end were answered using justification provided from the output of the program. If the program didn't work as expected, the issue is discussed in the writer's answer. The written answers only refer to numbers and plots that are shown in the notebook.</td>
<td>Never</td>
</tr>
</tbody>
</table>

**Overall feedback**

[Feedback for the author](#)
Weeks 1-8: Peer review using Moodle “workshop”

Teacher post-assessment view:

Students see individual feedback and grade for submission and assessment.
Lessons Learned

1. Need to know how they will use skills in future - panel of experts in lecture
2. Need incentives to come prepared - post-lab review quizzes
3. Need early & frequent positive feedback - embedded peer review
4. Students develop proficiency but find process uncomfortable!
Impacts on Attitudes

On aggregate: small drop on all measures; only significant change is usefulness
Not all bad news...

Confidence

Pre-Survey
Post-Survey

Interest

More students at high tails of distribution...
More students at high tails of distribution – especially amongst women!
1. Need to know how they will use skills in future - panel of experts in lecture

2. Need incentives to come prepared - post-lab review quizzes

3. Need early & frequent positive feedback - embedded peer review

4. Students develop proficiency but find process uncomfortable - how can we continue to improve their confidence?

All notebooks available online: jennyfisher.github.io/computing-modelling-earthsci

Thanks to B. Bukosa, J. Greenslade, S. Inakollu, N. Page, & EESC102 students!

Jenny A. Fisher, jennyf@uow.edu.au, @AtmosJennyF