POGIL in Computer Science II: An Update  http://cspogil.org

Clif Kussmaul. Associate Professor of Computer Science, Muhlenberg College
Dan Libby. Professor of Chemistry; Carl Salter, Professor of Chemistry, Moravian College
kussmaul@muhlenberg.edu, rdlibby@moravian.edu, csalter@moravian.edu

Background
In Spring 2011, the authors received a 3 year NSF TUES grant to develop, validate, & disseminate sets of POGIL activities for CS; and next, to foster a POGIL community within CS.

Key components include:
- Activities developed & piloted by CS faculty with POGIL experience.
- Reviewed by 2 chemistry faculty with extensive POGIL experience.
- Reviewed & tested by CS faculty with POGIL workshop training.
- Consulting from assessment expert.
- Dissemination via workshops, etc.

Table 1: Project Timeline (PA=POGIL activity)

<table>
<thead>
<tr>
<th>When</th>
<th>PIs &amp; co-PIs</th>
<th>CS Collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011 Summer</td>
<td>plan assessment</td>
<td>POGIL training</td>
</tr>
<tr>
<td>2011-12 Year</td>
<td>create/refine PAs</td>
<td>review/PAs</td>
</tr>
<tr>
<td>2012 Summer</td>
<td>create/refine PAs</td>
<td>POGIL training</td>
</tr>
<tr>
<td>2012-13 Year</td>
<td>refine PAs</td>
<td>review/refine PAs</td>
</tr>
<tr>
<td>2013 Summer</td>
<td>create/refine PAs</td>
<td>create/refine PAs</td>
</tr>
<tr>
<td>2013-14 Year</td>
<td>refine PAs</td>
<td>eval (treatment)</td>
</tr>
</tbody>
</table>

Status
To date, we have drafted, piloted, & revised POGIL activities for:

Programming
1. Unit Testing (JUnit)
2. Error Handling & Exceptions
3. File Input & Output
4. Object-Oriented Design
5. Object-Oriented Inheritance

Data Structures & Algorithms
6. Searching (linear & binary)
7. Queues
8. Stacks
9. Linked Lists
10. Maps & Hash Tables
11. Sorting (mergesort & quicksort)
12. Bioinformatic Sequence Analysis

Software Engineering
13. Teams & Roles (introductory)
14. Software Life Cycles
15. Risk Management
16. Scheduling (PERT, critical path)
17. UML Analysis Diagrams
18. UML Design Diagrams
19. Task Tracking
20. Version Control

Table 2: Project Assessment

<table>
<thead>
<tr>
<th>Goal: Enhance…</th>
<th>Assessed using…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning outcomes</td>
<td>Grade distributions, Qualitative assessment of some assignments.</td>
</tr>
<tr>
<td>Affective outcomes</td>
<td>Current (e.g. SIR-II), existing (e.g. TDS), &amp; custom instruments.</td>
</tr>
<tr>
<td>Recruiting &amp; retention</td>
<td>Course enrollments &amp; major/minor counts.</td>
</tr>
<tr>
<td>Quality of PAs</td>
<td>Activity reports, peer review, interviews.</td>
</tr>
<tr>
<td>Affective outcomes</td>
<td>Reflection, interviews.</td>
</tr>
<tr>
<td>Tech (FOSS) to support PAs</td>
<td>Activity reports, peer review, interviews.</td>
</tr>
<tr>
<td>Awareness (talks/papers/workshops)</td>
<td>Activity reports, interviews.</td>
</tr>
<tr>
<td>CS-POGIL community</td>
<td>Attendance &amp; evaluation forms.</td>
</tr>
</tbody>
</table>

POGIL fits CS well, since CS students must master fundamental concepts; learn to work in teams; and learn how to learn new theories & tools. POGIL in CS presents opportunities:
- Explore new topics(s) & discipline(s).
- Explore concepts & techniques that span disciplinary boundaries.
- Produce tools for the broader POGIL community.

POGIL in CS also presents challenges:
- Few activities, emerging community.
- Content changes & evolves quickly so developing activities is difficult.
- Curricula vary widely – intro courses use varied approaches & languages, making reuse difficult.

Future plans include more activities, and fostering a broader community of educators in CS & other areas.

Acknowledgements
To NSF for TUES grant DUE-1044679, and to US-India Educational Foundation for a Fulbright-Nehru award.

Sample References