

TEACHING PHILOSOPHY

W. Randolph Franklin

ECSE Dept, 6003 JEC, Rensselaer Polytechnic Institute, 110 8th St, Troy NY 12180

(518) 276-6077, fax: (ask)

frankwr@rpi.edu, mail@wrfranklin.org or wrf@ecse.rpi.edu

<http://wrfranklin.org>

January 31, 2007

A useful teaching philosophy must align with the real world. First, students need to have an intellectually well grounded program, and not just learn how to use the current commercially popular tools. For instance, there are general principles of operating systems that transcend particular operating systems. We're being paid for our expertise; and we must resist potential employers' demands to teach what they want.

However, the program still needs to be connected to applications. Since before the Renaissance, science and engineering have been energized by the task of solving important applications. The students need to graduate with practical problem-solving skills. The necessary skills range from the ability to use the major available tools, to a knowledge of their limits, to a lifelong desire to learn new paradigms as they become practical.

Current events provide the most compelling applications. A course cannot be designed only once and then frozen. It must be refreshed to be relevant. When teaching a course related to my research, I use my experience to broaden the students' horizons. Whenever a computer engineering problem made the news when I was teaching the required sophomore course Computer Components & Operations, I described the technical details to the class, formulated so that they could understand it. One example was the file system overflow on *Spirit*, the first Mars Rover, which caused repeated rebooting, and which was fixed with a low level patch.

The students must also see examples of successful people, whom they may emulate. While teaching the senior course Computer Graphics, I have created a web site listing successful RPI alumni related to Computer Graphics, at <http://www.ecse.rpi.edu/Homepages/wrf/pmwiki/Main/FamousAlumni>

After learning specific facts, and how to reason about them, students should learn how to learn on their own. The best undergrads might practice this with an undergrad research experience.

All strategies require well-ordered tactics, so here are some, occasionally controversial, proposals.

1. Active NSF PIs can use the NSF *Research Experiences for Undergrads (REU)* program (NSF-02-136) to support one or two undergrads on research each summer. Every NSF PI should be requesting REU supplements each year. When at NSF, I approved all such requests. Now back at RPI, I request annual REU supplements from NSF. In addition, this past summer, I also funded two more undergrads from my general research funds.
2. We should resist the urge to be original, merely for the sake of originality. When I teach a course for

the first time, I try to copy others' material. In turn, the material from the courses I've taught has been available on my website since 1993 for free downloading by anyone on the web.

3. Nevertheless, we should not force a standard solution where it is inapplicable. E.g., computer science majors and information technology majors might arguably be better served with separate courses. When I was an undergrad at Toronto, I much preferred the subjects with more targeted courses. There were 5 levels of freshman math, 3 of freshman physics, and 1 of freshman chemistry. I never took another chemistry course, but took much more math and physics.
4. When appropriate, the course study techniques should match the real world. E.g.₁, since reference material is available in the real world, my exams are open book. E.g.₂, since people collaborate in the real world, I allow collaboration on homeworks and projects (but not on exams). E.g.₃, since real world work is incremental, I allow students to build term projects on existing work, provided that they have permission and provide an acknowledgement.
5. The students deserve face time with the profs. Professors should lecture even in the freshman courses. We should use the support staff, such as grad students, behind the scenes. I have lectured in classes of all levels and sizes. I use TAs for grading and labs. Indeed, face time with our intelligent and creative students is reinvigorating. Also, I encourage questions about Computer Science or engineering or RPI in general, and encourage contacts from former students. Finally, I don't pretend to know more than I do, but encourage students to point out my mistakes. Trying to find mistakes, helps to keeps them awake in lectures.
6. Students should be treated as professional colleagues. When a student reports (in a timely manner) that we lost a homework, I automatically believe him/her, at least the first time. Some students abuse this; others feel like adults.
7. Students require that course mechanics run smoothly. E.g., I cover each important topic at least twice in class, and list it on a lecture summary. Homeworks track the lectures; exams track the homeworks.
8. The undergrad experience would improve if colleges treated large courses as profit centers allowed to reinvest in themselves, instead of as cash cows to support the graduate courses. Then the large courses would have the resources to improve even further, perhaps eventually producing patented products.

I have tried to apply the above points in my teaching since 1978. I've taught everything from freshman intro programming courses to grad courses. I've taught and developed courses both related and unrelated to my background. E.g., although my background is computer science, math, and physics, I've taught Computer Components & Operations, the intro sophomore EE and computer engineering course several times, and also extensively revised and helped convert it to a studio format, and then back again.

Universities must break even financially. The gross tuition in my courses, after prorating for shared courses, is often over ten times my salary. My senior ECSE Computer Graphics course, although not a required course, was sometimes the largest senior course in ECSE, while simultaneously receiving very high student scores. That demonstrates my style's effectiveness.

Finally, I'm familiar with teaching at other institutions for these reasons:

1. having been a CSAB or ABET site visitor five times, and
2. being on the Union College (Schenectady NY) Computer Engineering Program external advisory committee.