

# UPSILON PI EPSILON

Jeffrey L. Popyack

## Are You Ready for the Renaissance?

### A RENAISSANCE IN COMPUTER SCIENCE? REALLY?

In a field that has been characterized since its inception by constant, dynamic change, it's difficult to imagine what a rebirth would mean. It has been equally difficult to spot a genuine trend until it's in full swing, so claims of an impending renaissance should be taken with a grain of salt. As an example, the declines in computing enrollments through the greater part of the last decade were clearly more than a momentary stumble. The apparent turnaround in recent years looks promising, but it is still too early to claim a full reversal.

In the interim however, a confluence of changes in the computing education landscape provides the foundation for a true renaissance. Whether to address declining enrollments, historically poor and decreasing female enrollments, inadequate preparation of entering students, misinformation and stereotypes about computing, etc., the spirit of these changes has been to embrace an observation credited to Henry J. Kaiser: "Problems are only opportunities in work clothes." At this point, it is exciting to see the hard work bearing some fruit, most recently the congressional declaration of National Computer Science Education Week. As we gaze into the crystal ball and ponder the possibilities and opportunities presented, a vision of a very different future emerges.

### Changes in the Pipeline

Consider these changes in secondary education, as we begin to imagine what college computer science education will be like in the next decade. Perhaps the most visible and jolting recent event was the decision in 2008 by the College Board to discontinue the Computer Science AB exam. In itself, this would be upsetting, and yet when taken in combination with these other activities, can hardly be seen as anything but an opportunity. To name a few:

- The Computer Science Teachers Association's work to establish a national K-12 computer science curriculum, and formation of local chapters of CSTA;
- Use of Alice, Scratch, and other environments for learning important concepts of programming and computational thinking through storytelling and animation in K-16 curricula;
- Increasing popularity of FIRST Robotics for high school students and introduction of FIRST Lego League for younger students. While these programs were introduced in the 1990's, they began reaching formidable numbers in the last five years.

This certainly isn't what they were teaching when I was in high school,

and it probably doesn't resemble what you saw, either. Potential students' perceptions about computing education are changing, but the real question is whether those perceptions are any closer to reality than they used to be. Put more bluntly: is their mental image of the possibilities of computing a more accurate portrayal of computing at the college level, or of something beyond college? How should the ever-changing computer science curriculum adapt?

### Changes Closer to Home

In 2007, The National Science Foundation introduced the CISE Pathways to Revitalized Undergraduate Education in Computing (CPATH) program, in large part to help

---

**Perhaps the most visible and jolting recent event was the decision in 2008 by the College Board to discontinue the Computer Science AB exam.**

---

rethink the way core computing concepts, methods, technologies, and tools may be used beyond the traditional computer science curriculum. The first focus was on "Community Building" as a way of expanding the discussion beyond CS educators and into an increasingly larger community. And in 2009, ABET's Computing Accreditation Commission made some interesting revisions, among them removing explicit requirements for differential and integral calculus, probability and statistics, a full year sequence in a laboratory science, and even for explicit numbers of mathematics and science courses (while retaining the total amount of math

## Upsilon Pi Epsilon

*continued*

and science credits at one year). How might these somewhat disparate activities combine with changes at the secondary level to form the perfect storm for radical revision in computing education?

### Make Way For the Renaissance!

Like its (to date) better-known counterpart which began in the 14th Century, the Computer Science Renaissance should influence not only science and engineering endeavors, but likewise art, politics, philosophy, literature and indeed all intellectual pursuits. How does your curriculum do this now? Students in a liberal arts curriculum are likely well grounded in such diverse fields. However, how well is computing integrated into those other curricula?

Let's start with mathematics and science: if your program has been accredited by CSAB/ABET, you now find yourself with a puzzling conundrum: require your students to take the same amount of mathematics and science, yet not necessarily the same courses. You may find that the choices are moot. Yet, perhaps not – this is an opportunity to work with mathematics, science, and engineering faculty to develop courses with components decidedly rooted in computational thinking.

Next up is art: I don't know about you, but I have usually found the overlap between students drawn to technical disciplines such as computer science and those interested in creative arts to be small. Yet certainly, the overlap is there, even at a technological university like my own. A Bachelor of Arts degree provides a mechanism for students to pursue dual interests, but in most cases, those interests operate independently.

Two recently launched programs that have caught my attention go a step further. At the University of Virginia, the Bachelor of Arts in Computer Science was established in 2006 with an interesting twist: whereas the department is housed in the College of Engineering and Applied Sciences and the BS/CS is offered as an Engineering degree, the BA/CS is offered by the College of Arts & Sciences – with plenty of participation from the CS Department. At Carnegie Mellon

University, where interdisciplinary degree programs abound, they established a Bachelor of Computer Science and Arts (BCSA) degree. This is far from a standard BA in CS, however, combining a selected concentration from the College of Fine Arts with a CS concentration.

Noteworthy in both cases is that the computer science department is cooperating with a unit in another college

interested in patent law or intellectual property law, a bachelor's degree in science/technology is essential. In order to even be allowed to take the Patent Bar Exam, the candidate must demonstrate satisfactory educational training in science and technology. Having a bachelor's degree in computer science from a CSAB/ABET-CAC accredited program (provided it was accredited on or before the date

---

## The paucity of scientists and engineers in Congress is regularly lamented. Depending on how you count, US Congress members with educational backgrounds in science/technology number from 2 to 5 percent.

---

to provide an interdisciplinary degree program. This could easily become the model for future academic programs featuring a computational component.

Onward. Being the gentleman I am, I will refrain from discussing sex and religion here. But, politics? The paucity of scientists and engineers in Congress is regularly lamented. Depending on how you count, US Congress members with educational backgrounds in science/technology number from 2 to 5 percent. The number of computer scientists is approximately zero. In any case, there's quite a gap before you reach the percentage of lawyers, which hovers in the 40-50% range. This being a renaissance, however, we should not be asking if you're a computer scientist or a lawyer – we should wonder why more computer scientists aren't lawyers! I have learned a lot about pre-law programs recently as our program considers a pre-law option for CS majors. For instance, while an overwhelming number of law school applicants have degrees in political science, roughly 0.3% of applicants have a degree in computer science. Still, if you're

the degree was awarded) is sufficient.

Considering these examples as examples, where else might a significant component of computational thinking be warranted? Indeed, where would it not?

### In Conclusion

As computer scientists, we have long noted how central the concepts of computing are — or should be — throughout college curricula and indeed, society at large. And yet, the manner in which they show up has been uneven at best. The challenge to us as CS educators is to provide the foundation to instill computational thinking at all levels. But how, you may ask, can you convince your neighbors to listen? Tell them the Renaissance sent you! **Ir**



**Jeffrey L. Popyack**  
Dept Math and Computer Science  
Drexel University  
Philadelphia, Pennsylvania 19104 USA  
[JPopyack@drexel.edu](mailto:JPopyack@drexel.edu)

DOI: 10.1145/1721933.1721946

Copyright held by author.