William E. Byrd—Teaching Statement

---

1 Experience

I have extensive experience teaching computer science, having taught for fifteen semesters at Indiana University, both as a course instructor and as an associate instructor. I co-wrote *The Reasoned Schemer* (MIT Press, 2005), one of two texts used in IU's introductory undergraduate (C311) and graduate (B521) programming languages courses. I am currently supervising several undergraduate projects by former students, including research on neurosymbolic computation and development of a rule-based language for interactive fiction. I am a former public school teacher, and the former director of a special needs summer camp. I have organized tutoring and mentoring programs at both regular and special needs schools, and have taught everything from recursion and advanced logic programming to model rocketry and how to properly clean a latrine. Once I was asked to teach stand up comedy to a troop of Girl Scouts.

2 Teaching Philosophy

I have been privileged to work with more than my fair share of master teachers, and over many years have developed a personal teaching philosophy based on the following:

- The importance of continuing education for a teacher.
  It is a student's right to be taught by someone with real expertise in their subject.
- The importance of properly motivating students.
  Learning something interesting or surprising is inherently motivating.
- The importance of projects with universal appeal that students can show off to friends, parents, and grandparents: fractals, animatronics, digital fabrication/3D printing, computer-controlled musical instruments, etc. *Bling is good, bling-bling is better.*
- The importance of ambitious projects that push the skills and creativity of students. *We will succeed epically or fail epically.*
- The importance of embracing mistakes, and of ensuring that students have enough chances to make mistakes. *An expert is someone who has made more mistakes than you have.*
- The importance of semi-structured activities in which students can learn from each other. *In the words of FIRST robotics hero Woodie Flowers, "Learning with always trumps learning from."*

I pride myself on getting to know my students personally. At the beginning of every semester I invite my students to lunch in small groups to learn about their interests and experiences. I use this information to specialize the course around their common interests. For Indiana's honors introductory programming course (H211) in Fall 2010, almost every student was interested in games; as a result, I based all but one of our projects on games. (The remaining project was fractal

movie generation, which was also extremely popular.) The following Fall my H211 students were less interested in games, but almost every student played a musical instrument. For our projects we procedurally generated music using Markov models and built Arduino-controlled polyphonic touch-sensitive electronic pianos; my students' capstone projects included an Arduino-controlled robotic flute that uses compressed air and servo motors, a glove-based synthesizer that automatically transcribes the music being played, and a 3D-printed glowing jellyfish that "dances" to the beat.

3 Arduino, Digital Fabrication, and CS Education

I have become fascinated with how digital fabrication, 3D printing, and Arduino can be used to help students learn programming and computer science in new ways. After two years of integrating Arduino and physical computing into Indiana's honors introductory programming course (H211), the School of Informatics and Computing asked me to design and teach a new course on Arduino. I also helped with the electronic textiles workshop at the 2010 Indiana Celebration of Women in Computing (InWIC), and helped two of my former H211 students, Maria Khokhar and Brittany Moore, plan a 3D printing demo which won the "People's Choice" award at InWIC 2012. Maria and I are building an Arduino-controlled Segway clone (a "Seg-bot"), which we work on whenever I visit Bloomington. We took a welding course together so we could build the Seg-bot frames.

With the help of my former H211 students, I started informal Arduino/physical computing and 3D printing clubs, which are now official student-run activities. Club members build and operate Open Source Arduino-controlled 3D printers, design and model printable musical instruments, and control those instruments using servo motors and Arduino. In 2011 Mark French, a professor in Purdue's Mechanical Engineering Technology (MET) Department, invited me to speak about Open Source 3D printing in October; based on my visit, the MET Department created an undergraduate digital fabrication club and purchased kits for several 3D printers and computer-controlled routers.

During my trip to Purdue, Mark French and I visited an elementary school in Lafayette where I demonstrated 3D printing to a third-grade class; the students were mesmerized by the 3D modeling process, the operation of the printer, and the final printed objects. Most popular was the "cat soap" I made by 3D scanning my father's sculpture of our family's cat, 3D printing the model, vacuum-forming a mold using the printed piece, then filling the mold with melted soap. The girls especially loved the cat soap. I think there is a fantastic opportunity for undergraduate students to use digital fabrication technologies to show girls how science, engineering, and CS can be used creatively and artistically.

In 2011 I led a multi-week Arduino animatronics workshop at the Bloomington Boys & Girls Club. The workshop was a great success, and was especially popular with middle-school age girls. I am especially interested in teaching teens to use digital fabrication technology including 3D modeling and 3D printing—I would love to involve undergraduates in teaching digital fabrication at a local branch of the club.

4 Other Course Ideas

In addition to standard core CS courses, courses related to Arduino and digital fabrication, and specialized courses on relational/logic programming, functional programming, and programming languages, I would like to develop a "retro programming" course as a way to get students closer to the hardware. I would love to teach an Atari 2600 game programming course, somewhat similar

to the course Ian Bogost teaches at Georgia Tech but using 6502 assembly instead of a high-level language. Partly to prepare for such a class in the future, my friend Andy Keep and I have started a 6502 Retro Game Hacking Group at Utah.

I am also interested in how games can be used in CS education. I am currently advising one of my former H211 students, Brittany Moore, on the design of Weaver (`https://github.com/brittanymoore/weaver`), a declarative programming language for Interactive Fiction (IF) based on defeasible logic. In addition to involving other students in this research, I would like to develop a course on IF. This course would appeal to students who enjoy writing and telling stories.

As both a computer scientist and a gamer, I am fascinated by the StarCraft family of real-time strategy games, which have been described as "chess for the highly coordinated." I would like to form a student team to enter the annual AIIDE StarCraft AI Competition (`http://skatgame.net/mburo/sc2011/`). UC Berkeley has used this competition to teach AI to undergraduates and as a fertile ground for research, leading to the Berkeley Overmind project (`http://overmind.cs.berkeley.edu/`). If this experiment is a success, it could lead to a course that uses StarCraft to teach various ideas from AI—for example, students could create a build-order optimizer using genetic algorithms or constraint logic programming.