

“...But What Do the Top-Rated Schools Do?”

A Survey of Introductory Computer Science Curricula

Jeffrey Forbes
Duke University

forbes@cs.duke.edu

Daniel D. Garcia
University of California, Berkeley

ddgarcia@cs.berkeley.edu

SUMMARY

Our field continues to be blessed (and plagued) with continual curriculum change, from languages to techniques (objects first) to perspectives (“sage on the stage” vs. “guide on the side”). Particular emphasis has been spent crafting and re-crafting our introductory curricula [1]. This makes sense, since that not only defines the foundation upon which our upper-division courses are based, but is exactly where we attract (or lose) our best students who had not considered majoring in computer science. With enrollments declining, retaining our fence-sitting prospective majors takes on that much more importance.

When considering curriculum change at the introductory level, it often helps to look around at successful programs to see what they do. We surveyed the schools whose computer science Ph.D. programs were listed as the top 30 by the 2007 U.S. News & World Report ranking [2].¹

While other surveys have focused on departments, salaries and degree production [3], ours tried to capture the important aspects of each institution’s lower-division curriculum. First, we looked at the material on each department’s web page and course pages when accessible. We followed that initial sweep by asking representative faculty at each institution to report on the following questions, divided into seven major categories:

Institution

- Is the institution on quarters or semesters?
- Are the classes taught every semester? In the summer?
- What are the introductory, lower-division courses, and how do they map (if at all) to the standard CS0, CS1, CS2, etc?
- Is a literacy course available for students not interested in programming?

¹ We realize this survey is skewed to large research universities and ignores smaller teaching colleges, many with outstanding undergraduate programs. Perhaps a follow-up survey of the curricula at those schools (by interested parties) could yield interesting comparisons and contrasts?

- What is the flexibility in the lower-division sequence?
- Are the courses part of a “common-first-year”?
- Is ethics taught in the introductory sequence?
- Is there a “survey” course available?

Staff

- What staff is required to teach each course (faculty, teaching assistants, readers, lab assistants)?
- Are the instructors research faculty, teaching faculty, or graduate students?
- Are the courses taught by the same person every semester?
- Who teaches discussion sections / recitations? Labs?

Demographics

- What are recent enrollment numbers?
- How much have those numbers dropped (if at all) recently?
- What are the drop / withdraw / failure rates?
- What is the demographic of the student body?
- Are any non-majors required to take the courses?
- What is the typical grade histogram for the course?

Content

- What versions of what languages are taught?
- What are the textbooks used?
- Is there a feeling that the course is “fresh” or “stale”?
- When was the last major course facelift?
- What works really well and what is broken?
- What overhauls (if any) are planned for the courses?

Delivery

- How many contact hours are there and what is the breakdown into lecture / lab / discussion?
- How are the labs run? (instructor-led vs. problem-driven)?

Style

- What programming paradigms are covered?
- Is the first introduction to OOP an objects-first approach?
- Is there any pair programming (or other XP techniques)?

Meta

- Is there an institutional or departmental grading guideline, e.g., that the average GPA needs to fall in a certain range, or that there should be a given distribution of As, Bs, etc.?
- What are the innovative techniques being deployed, e.g., multimedia data as first-class objects, a collaborative content-delivery system, active learning, graphics, etc.?
- What questions are missing on this survey?
- Was there anything not captured by this survey you would like to add about your institution, staff, demographics, content, delivery or style?

We found that the schools used a large variety of approaches, but there were some common themes and clusters that emerged. For example, the first courses had almost as many different textbooks as schools. This model is distinct from some upper-division courses, (e.g., Artificial Intelligence, Algorithms, or Graphics) in which most schools used the same reference textbook. We will present the common and unusual cases, as well as celebrate the innovation that is taking place at various institutions.

Categories and Subject Descriptors

D 3.2 [Programming Languages] Language Classifications

K.3.2 [Computer and Information Science Education]: Computer Science Education, Curriculum

General Terms

Languages, Theory

Keywords

Computing curricula, computer engineering curriculum, computer science curriculum, information systems curriculum, information technology curriculum, software engineering curriculum, comparison of curricula.

1. OBJECTIVE

The objective of this special session is to have the opportunity to discuss the wide range of results we encountered after running the extensive survey. As it often takes a full *twenty minutes* in a paper session to describe a *single* innovative program, course, or approach, we feel a special session is warranted to do justice to presenting all of the results, describing the clusters and outliers from the analysis, and spotlighting the exceptional programs.

2. SESSION OUTLINE

Attendees will be presented with a handout of all the results we tabulated. We will begin the session by listing the institutions we surveyed, as well as presenting all the questions we asked. For every question, we will show the range of answers and rough clusters that emerged. As we proceed through the questions, we will make the discussion interactive by asking the audience to

estimate the distributions. After polling the attendees, we will show the histogram data of the actual results. We will also take time to highlight any surprising results.

When we have finished with the responses to the individual questions, we will discuss the groupings that emerged along different dimensions that could yield promising faculty collaborations. For example: *Schools A, B, C and D essentially follow the same three-course sequence, except school B just switched to Java from C++ to teach data structures.* We will also celebrate unique approaches and best practices we observed, and close with future directions. Adequate time will be left to address questions and receive feedback.

3. EXPECTATIONS

This session is targeted for any instructor who is interested in discussing approaches to introductory computer science. Many SIGCSE attendees will be among those surveyed.

The surveyed departments are of particular interest to computer science educators. The top graduate programs produce a significant portion of the future computer science educators and by virtue of size and visibility, often have a great deal of influence over the adoption and proliferation of curricular programs. By considering the manner that introductory computer science is taught at these institutions, we can observe and critique some overall trends in the field. Further, these schools have a variety of different approaches, and understanding the taxonomy and best practices could yield curricular insight.

One of the aims of this session is to solicit feedback from attendees, before, during, and after the conference. We have established a Wiki for this survey [4]. We hope that participants will add their comments, corrections and additions to the site, so that this discussion of teaching practices can continue after the conference.

4. ACKNOWLEDGMENTS

Thanks to Brandon Johnson, Beth Trushkowsky, and Jeremy Huddleston who helped create the survey and compile and analyze the data. Thanks to all of the faculty, staff and departments that provided the course information that made this survey possible.

5. REFERENCES

- [1] The Joint Task Force on Computing Curricula, "Computing Curricula 2001," *Journal on Educational Resources in Computing*, 1:3es, September 2001.
- [2] "America's Best Graduate Schools 2007", U. S. News & World Report, Washington, D.C., 2006.
- [3] 2004-2005 Taulbee Survey, Computing Research News, 18:3, May 2006.
- [4] <http://www.cs.duke.edu/csed/openwiki/doku.php?id=teaching:start>