Bringing CCIS to the Top

EXECUTIVE SUMMARY

Nick Flower

ARTICULATE INSIGHTS | WWW.ARTICULATEINSIGHTS.COM

BRINGING CCIS TO THE TOP

DOCUMENT #2

EXECUTIVE SUMMARY

The computer science school is facing a huge amount of competition from online courses, self-taught people, and coding bootcamps.

Although the bootcamps are often derided by the academic institutions that they are threatening, there are noteworthy high profile endorsements (President Obama https://www.whitehouse.gov/the-press-office/2015/03/09/fact-sheet-president-obama-launches-new-techhire-initiative, Peter Thiel https://www.edsurge.com/n/2015-01-13-thiel-backed-learn-to-code-startup-thinkful-raises-4-25m-series-a,)

Beware Peter Diamandis' '6 Ds of Exponentials', which illustrates an innovations rise from obscurity, to take over, to mundanity. In this chart, bootcamps, much like online courses, are in the 'deceptive' stage, where it looks like they won't work, and will never work. With enough time and effort however, they may very well pass into the 'disruptive' stage, where they take over everything. In 2015, 16,000 students are projected to graduate from a coding bootcamp, with courses costing around $11,000. For the price of half a semester at Northeastern, a student may find themselves just as well prepared as a Northeastern graduate. (http://www.bloomberg.com/news/articles/2015-06-08/coding-boot-camp-enrollment-soars-as-students-seek-tech-jobs) It's possible that bootcamps will not take off, be proven ineffective, and the unaltered computer science curriculum will live to see a glorious ten more years without having to adapt.

But what if that's the wrong bet, and the School of Computer and Information Science ends up on the wrong side of history?

The College of Computer and Information Sciences should consider the following:

1. Restructure the Introduction to Computer Science course to take advantage of the way that computer science has progressed in recent years, with a focus on doing scientific testing on the environment and the libraries used
2. Replace Scheme and Racket with more modern languages like Python or JavaScript
3. Invest in the creation of a Survey of Mathematics course which aims to educate students on how to identify when a real problem belongs to a certain branch of mathematics, as well as the core concepts in those branches
4. Redo the Theory of Computation course or make it non-mandatory
5. Restructure the Programming Languages course
6. Create an industry outreach program designed to attract top industry talent to professorship roles

We will look into each of the above recommendations in detail, with the reasoning of each section explained as clearly as possible. The goal is to multi-fold: To keep the school from falling into obscurity or otherwise losing its stature, and to make it, and the university at large, a pioneering institution, able to witness and adapt to the changing tides the future brings. To this end, it is worth re-evaluating the computer science curriculum and the courses it is composed of.