Pander to Ponder

Owen Astrachan
Department of Computer Science
Duke University, Durham, NC 27708
ola@cs.duke.edu

ABSTRACT

Ponder means "to weigh in the mind with thoroughness and care" [31]. Pander means "to cater to the weaknesses and base desires of others" [31]. We report on a course we have designed and delivered over a six year period. The course was originally designed as a technical writing course for majors, but has evolved into a non-major's version whose enrollment ranks it as one of the three most highly-enrolled and thus arguably most popular courses for undergraduates at our university. We have worked diligently to ensure that students ponder the topics and problems that comprise the material for the course — and the material is deeply technical at many levels. We have also pandered to student needs in meeting curriculum requirements, offering the course at a time convenient for athletes and others, and using popular media when possible. We started with the goal of engendering interest and passion for computer science and how it affects the world. We report on our efforts to attain this goal while keeping material appropriately technical. We claim our students are engaged in a different type of computational thinking than that espoused in [32, 5, 15]. For the purposes of this paper and discussion we call our approach *pander-to-ponder*. We provide examples and illustrations of the material we cover, relate it to similar courses at other institutions, and show how we use problems to motivate learning. In the work we report on here the learning is specific to understanding how contributions from computer science are changing the world.

Categories and Subject Descriptors K.3.2 [**Computers & Education**]: Computer & Information Science Education — *Computer Science Education*

General Terms Legal Aspects, Human Factors

Keywords Computational Thinking, Problem-centric Learning, Non-majors, Ethics, Computers and Society

1. INTRODUCTION

Marc Snir [28] writes "[Computer and Information Science] will need to address the cognitive, social, economic, legal or ethical constraints in the design of digital systems; increasingly it will need to think of the human or social group as an integral part of [the] system]." This is very close to the view we have taken in the design and implementation of our courses. However, we are trying to inculcate in our students an understanding of the social, economic, legal and ethical constraints *that are part of the Internet and its related components*. We do want to discuss the constraints in the design of such systems, but we want to go beyond design to study constraints in the use, deployment, governance of the Internet, and the changes engendered by wide-spread use of the Internet. We provide the original rationale for the development of the course and its evolution into the current form on which we report here. We then offer comparisons and examples that illustrate how our course is delivered, what students do, and how this is similar to and different than related courses at other institutions.

2. DEVELOPMENT OF THE COURSE

Our course was originally developed as a major's elective that fulfilled a new university *writing-in-the-discipline* requirement. We have taught this course once a year since 2002. We envision our course and refer to it as IP^2: the Internet Protocol meets Intellectual Property. We return to the idea of two views from different constituencies in explaining the development of the course below. We have often relied on materials from [1] in developing our course, though both the audiences and courses are very different.

2.1 Population and Demographics

Duke is a private university with approximately 6,400 undergraduates. There are two schools for undergraduates: Engineering with 1,100 and Arts and Science with the remainder; the department of Computer Science is in the school of Arts and Science. In 2002 approximately 125 students graduated with a degree in Computer Science compared to approximately 40 students in 2008; we offer both an AB degree and a BS degree in computer science. Student demographics in the major's course we discuss in this paper reflects the major: roughly 20% of the students in the course are women in each semester in which the course has been offered. The gender breakdown in the non-major's version of the course reflects the overall student population: just more than 50% of the students in the course are women.

2.2 Course Details

In its first offering, on the tail-end of the enrollment boom in computer science, we had three seminar sections of 15

SIGCSE '09, March 4–7, 2009, Chattanooga, TN, USA.

majors. In 2006, the last time we offered the course exclusively to majors we had one section of 15. We teach in a research university, our school is typically ranked in the top ten undergraduate universities in the United States. After offering a major's and a non-major's seminar in 2006, we began to see an increase in demand from non-majors. We reasoned that by removing the writing component, changing from a seminar to an active-lecture format, applying for the course to meet a university-wide requirement in ethics (which arguably it had met from its inception), and moving the time at which the course was offered, we could likely dramatically increase the enrollment.[1] Although we cannot be certain which of the changes led to the increase we are experiencing, we are now teaching the course to more than 230 students in one section this fall. Our programming courses for non-majors have also seen a dramatic increase in enrollment in the past year (from 60 to 100), but we cannot continue to scale those courses to meet demand since they require daily hands-on lab activities.

Our course first used the original version of Lessig's *Code and Other Laws of Cyberspace* [18], now available in a second edition and online, as the basis for its development. The important metaphor from that book we embraced is that there are two kinds of code that govern the Internet: west coast [computer] code from Silicon Valley, and east coast code [of law] from Washington DC. The *regulability* of the Internet by both kinds of code is an important foundation of our course. The potential for regulability is due in large part to the end-to-end design principles [25] that were foundational in the design of the Internet. These principles and their repercussions are explained well in a computer science paper *Tussle in Cyberspace* [6] we use to illustrate the similarities and differences between how the Internet is viewed by authors from different fields. We continue to embrace this code-metaphor using a new book *The Future of the Internet And How To Stop It* [34] that is available for purchase and online.[2] This book provides an insightful view and the opportunity to see the author interviewed on the *Colbert Report* [8] in discussing the book and the Internet. We return to this interview later, but the availability of such interviews is part of what we argue is the basis for the potential success of our *pander-to-ponder* approach to course development.

3. PROBLEMS AND STORIES

In developing our seminar course into a large, active-lecture format we relied on examples and philosophies relating storytelling to the learning of mathematics and computer science [3, 23] and to learning in general [26]. We choose our stories from current events and from those that have worked well in previous versions of this course. We also use stories relayed by our friends as part of daily living and coping with the Internet and technology in general. In the next sections we illustrate this storytelling *pander-to-ponder* approach by several examples, relating these examples to computational thinking, but showing how our approach differs from what is typically espoused in courses emphasizing computer science rather than the changes facilitated by computer science. We return to these differences as we discuss three of the stories we use in developing reading and materials for our course.[3]

3.1 P2P, Algorithms, Business Models

The rise of smart phones in general and the iPhone in particular has led to general interest in applications that run on cell phones. The business model of applications in the iPhone 2.0 release has engendered discussions, blogging, and serious writing at many levels. In our current offering of the course we surveyed all students at the beginning of the semester. Roughly 98% of the students in our course have a laptop, 90% have an iPod, and 15% of the students have an iPhone. One faculty member in our department mentioned in a lunch conversation the free iPhone application named *Shazam* built on technology likely requiring FFTs and other waveform techniques to identify songs in noisy environments by transmitting digital, audio fingerprints of the songs for identification [30]. In our class we asked if someone had an iPhone running Shazam. We anticipated the answer would be yes and were delighted that the "volunteer" sat far in the back of the room (more than 100 feet away from the speakers in the front). We played a song using the sound-system and over the voices of those filing into the classroom late. In fewer than 15 seconds the song had been identified. None of the students in the room knew the song. We then discussed both the technology and computer science behind how such a service works. Later in the semester we delve more deeply into how music has engendered widespread adoption and changes on the Internet due to file sharing, peer-to-peer networks, and copyright issues engendered by the DMCA in the United States. Students are often interested in these topics, but our introduction of the concept via a live demonstration, followed by a serious discussion of many of the issues, is part of the *pander-to-ponder* approach we have embraced. We will not ask students to understand the mathematics behind the FFT, but we will ask them questions whose proper answer requires understanding technologies and possibilities facilitated by the FFT. Similarly we will not ask questions about the details of Huffman coding, e.g., as done in the courses [19, 9]. Instead we will ask about the differences between lossy and lossless compression, how Bittorrent [7] uses techniques from economics combined with novel distributed algorithms and systems, and why the audio-identification tools offered by Shazam are free on the iPhone where it is marketed, but only available for purchase in the United Kingdom for other cell phones.

3.2 Encryption, and the Fifth Amendment

The case of *In Re Boucher* [21, 20] provides a wonderful illustration of the kind of story we look to find in current events that illustrates the technical and ethical issues we discuss in this course. To highlight the manner in which we use this court case to illustrate aspects of cryptography and encryption we first contrast coverage of cryptography as found in typical computer science courses, even those embracing a societal view. Then we provide a brief description of the court case before using it to explain our *pander-to-ponder* approach.

Typical explanations of encryption in the literature provide clear, but abstract explanations often beginning with a

[1] Here we see one of the pragmatic reasons for changing the format. The decrease in majors in our program has led to a sometimes not so subtle pressure from university-level administrators to increase enrollments in other courses.

[2] The online availability of this book, Lessig's book, and [12] provides a compelling example in understanding copyright, the Creative Commons, and open source licensing.

[3] All the materials for the course we teach are accessible from `http://compscipbl.org`.

story like "Bob wants to tell Alice a secret story" (the Bob and Alice explanations started in the original publication of the RSA public-key encryption system [24]). These stories do provide a simple explanation of encryption, but the stories lack strong motivation for non-major students. Public-key cryptography and encryption are a staple of courses for non-majors, especially those in courses emphasizing computational thinking [16, 10]. In notes for the course taught by Cortina [9] the explanation and assignments associated with cryptography and encryption emphasize large prime numbers, modular arithmetic, fast exponentiation, and the difficulty of factoring large numbers. These topics are, of course, essential for understanding how encryption and a public key infrastructure (PKI) work. The topics are also powerful examples of the contributions computer science makes to society. However, in our view assignments that emphasize the mathematical aspects of cryptography, e.g., those that require students to do math, do not help in motivating students to understand the power of the contributions computation makes. Assignments more similar to those we have begun to use are found in a Harvard course [19] that has led to a book [2] described in an audio interview with the Chronicle of Higher Education [14]. The Harvard course is informed in part by the work of Hal Abelson who has taught a higher-level course course at MIT, 6.805 [1]. Problems from the Harvard course require understanding *why* exponentiation and modular arithmetic must be used together in a public-key system to exchange keys. Rather than asking students to do arithmetic, the problems require that students understand why fast exponentiation and modular arithmetic are necessary.

We use an approach similar to what is espoused in the Harvard course, but we couch the development of cryptography and PKI in a current event that is compelling to students. Briefly, a Canadian (Boucher) traveling into the US is asked by the border police to show the contents of his laptop. The police find evidence of child pornography (claimed by the defendant as part of an accidental download). They confiscate the laptop, turn it off, and prosecute. However, the defendant has used PGP, *Pretty Good Privacy* to encrypt the contents of his laptop. When the police try to turn the laptop back on they cannot read its contents. The defendant uses the fifth amendment to claim he need not divulge the key to decrypt the laptop. Courts have upheld his claim, noting that he could be compelled to give up a physical key, but not the remembered combination to a safe. This case has been covered in the popular press [20] and leads to several interesting legal, ethical, technical and societal questions. We use the case as the basis for understanding PKI and encryption. We ask students to write answers to questions that show they understand some aspects of crypto-systems: "Why can't the border police crack the password?", "How does the development of PGP reflect open-source, patents, and Homeland Security issues". In the seminar-version of this course we held a mock-argument before the Supreme Court for this case. Students took on the role of Supreme Court justices, news reporters, and lawyers for each side in the case. Students viewed this as highly successful and we are working to develop an exercise that scales from 23 to 230.

3.3 Network Neutrality, TCP/IP, and BGP

Network Neutrality [33] can provide a deep area to explore the mix of technical, legal, political, and social aspects of Internet regulability. We can provide students with a technical examination [11] as well as a treatise examining the issues from an economics and legal point-of-view [27]. In each of the past three years we have used Network Neutrality as part of discussions in understanding both the architecture and the governance of the Internet. Students have written papers examining alternative points of view with respect to the different stances espoused by, for example, Tim Berners Lee and several members of congress. However, just as showing a video of Barack Obama discussing Bubblesort at Google [22] substantively changed the discussion of sorting in our CS2 course, showing videos of interviews on the Daily Show (with Jon Stewart) with congressmen and comedians discussing Net Neutrality [29] transformed our discussions and student analysis of Network Neutrality. This is a perfect example of the *pander-to-ponder* approach we report on. In this case we first pander to student interests, then we ask them to ponder the repercussions of a tiered Internet.

This discussion leads to how Net Neutrality would be enacted if businesses were able to offer tiered service. This requires understanding both how packets are routed and how the Internet is constructed from interconnected Autonomous Systems. In the past we focused discussions on the IETF and how RFCs, rough consensus, and running code have led to many Internet standards. Although we continue to study RFCs, we do so in conjunction with Zittrain's analogy [8] of routing as people-passing in a mosh pit and the story of how Pakistan inadvertently shut down world-wide access to YouTube [4]. Students appear interested and participate in class discussions when these stories are used to jump-start the discussions. Ultimately students will need to demonstrate an understanding of how BGP helps run the Internet, but we will not ask questions that test knowledge of how a dotted-quad is constructed as an IP address. We will, however, expect students to appreciate the humor in the bumper-sticker *There's no Place like 127.0.0.1*, to understand why it took two hours to shut down access to YouTube and only two minutes to restore access world-wide because of the transitive trust model explicit in BGP, and whether the Internet will stop working in 2012 if we do not change from IPv4 and 32-bit addresses to IPv6 and 64-bit addresses. We ask similar questions, but we develop understanding and interest using the *pander-to-ponder* approach.

3.4 Phishing and Social Experiments

All our students use the Internet on a daily basis. In our class of over 230 students in the fall of 2008, only two did not have a Facebook page. To the question "how does your view of the Internet differ from your parents' view" one student responded. "My parents view the Internet as an interesting addition to their daily lives, I view it as an integral part of my daily life." However, the students are not necessarily completely sophisticated in their understanding of phishing [17] and how it works. In class we discussed email and web phishing, so-called Nigerian/419 scams, and the CEO/CIO-based phishing attacks dubbed "whaling" in which executives receive fake subpoenas from the Justice Department. We reviewed the West Point [13] experiment in which student attention to phishing was part of e-security education. I also told the class they would be getting email that might be part of a phishing expedition for the course.

That day I sent email to everyone in the course from either dukeathlete@yahoo.com or dukechickee@yahoo.com. Students were told that the email came from a student in the course indicating that the student had discussed phishing with me and was forwarding a link to a website whose visitor-count would determine extra credit on the midterm. Authentication on the website was done through our standard university wide Shibboleth[4] system. However, the page students saw before authentication included a php script that generated the authentication link. Half the student visitors were taken to a fake authentication page and the other half to the real page. The fake page had no SSL lock visible in the browser and the URL was not the standard authentication URL. However, students gave their login-id and password regardless of which page they were taken to, at least as far as the logs of the page determine access. No passwords were transmitted or stored via the phishing site. Students were shown the code that generated the links, how simple it was to conduct the phishing expedition, and were given extra credit on the final. The exercise was a success both in understanding SSL, how scripts work and in validating the *pander-to-ponder* approach.

4. FUTURE DIRECTIONS AND SUMMARY

We anticipate that our students will likely not remember how to do exponentiation efficiently in understanding cryptography, why two's complement representation is used to represent numbers in binary, and what the format is in the header of a TCP/IP packet. However, we hope that students in our course will take away the passion we bring to the subject and might, in some cases, develop a similar passion. We hope that they will remember why a 40-bit encryption key is too weak and why Skype's 256-bit key is more than sufficient; we hope that they will understand why the original IPv4 guidelines for allocating IP addresses in chunks of size 8, 16, or 24 bits was vastly improved by Classless Inter-Domain Routing (CIDR) that scales more readily; we hope that they will understand the differences between IP filtering looking at packet-headers and deep packet inspection for copyright violations on a campus network. We work to ensure that students have understood the contributions that computer science has made to their lives and to society, from protecting their privacy (or endangering it) to ensuring that all votes are counted (or not) in each national election.

Just as visitors to wilderness areas are exhorted to "take only memories, leave only footprints", we want our students to remember our course, to remember their instructor, and to remember their school. We are fostering both a love of learning and a loyalty to an institution. This may seem crass, but we are not teaching this course to potential software engineers who we must meet national standards, who might develop mission or life critical code, or who may help build the infrastructure of a highly-successful startup. We are teaching this course to students who might develop a world-startling idea, motivated perhaps by the examples, stories, and understanding we work to develop in our course. We are teaching to future business and policy leaders. We want students to send email to us ten years after they take the course, to come visit at their twentieth reunion, and to tell stories about our course and our delivery of it to their friends, children, and co-workers.

[4]This is used by many universities, see shibboleth.internet2.edu for details.

It is our mission as educators to share our passion, not simply to develop competent and brilliant software designers. We have provided background and some foundation for how and what we have developed. Although the foundations of this course have been built over a six-year delivery of a seminar version, the large version is new this year. Rather than viewing this report as a premature examination, we hope it will serve as an example of how using tools and philosophies espoused by our colleagues has led to a new way of computationally thinking.

5. ACKNOWLEDGMENTS

We have benefited greatly from using resources made available from the 6.805 course at MIT [1], from conversations with graduate-student instructors who have taught sections in previous versions of the course, and from professors with whom we co-taught the course in previous years. This material is based upon work supported by the National Science Foundation under NSF grant 0722274. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

6. REFERENCES

[1] H. Abelson. Ethics and law on the electronic frontier. Accessed August 29, 2008, http://groups.csail.mit.edu/mac/classes/6.805, 2008.

[2] H. Abelson, K. Ledeen, and H. Lewis. *Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion*. Prentice Hall, 2008.

[3] O. Astrachan and P. Denning. Innovating our self image. In *The Papers of the Thirty-Ninth SIGCSE Technical Symposium on Computer Science Education*, pages 178–179, March 2008.

[4] R. Blog. Pakistan hijacks youtube. Accessed August 29, 2008, http://tinyurl.com/yuwd2p, 2008.

[5] A. Bundy. Computational thinking is pervasive. *Journal of Scientific and Practical Programming*, 1(2):67–69, 2007.

[6] D. Clark, J. Wroclawski, K. Sollins, and R. Braden. Tussle in cyberspace: defining tomorrow's Internet. *Networking, IEEE/ACM Transactions on*, 13(3):462–475, 2005. An earlier version appeared in the 2002 SIGCOMM Conference.

[7] B. Cohen. Incentives Build Robustness in BitTorrent. *Workshop on Economics of Peer-to-Peer Systems*, 6, 2003.

[8] S. Colbert. Colbert interviews Jonathan Zittrain. Accessed August 29, 2008, www.youtube.com/watch?v=hS8ywG5M_NQ, 2008.

[9] T. Cortina. http://www.cs.cmu.edu/~tcortina/15-105fa07/lectures.html, 2007. Accessed August 28, 2008.

[10] T. Cortina. An introduction to computer science for non-majors using principles of computation. *ACM SIGCSE Bulletin*, 39(1):218–222, 2007.

[11] J. Crowcroft. Net neutrality: the technical side of the debate: a white paper. *ACM SIGCOMM Computer Communication Review*, 37(1):49–56, 2007.

[12] R. Deibert, J. Palfrey, R. Rohozinski, and J. Zittrain, editors. *Access Denied: The Practice and Policy of*

Global Internet Filtering. MIT Press, 2008. also accessible online http://opennet.net/accessdenied.

[13] A. J. Ferguson. Fostering e-mail security awareness: The west point carronade. *EDUCASE Quarterly*, 28(1), 2005.

[14] A. Foster. Making sense of the 'digital explosion'. *Chronicle of Higher Education*, 54(49), 2008. Audio Interview, http://chronicle.com/media/audio/v54/i49/abelson_lewis.

[15] M. Guzdial. Paving the way for computational thinking. *Communications of the ACM*, 51(8):25–27, August 2008.

[16] P. Henderson, T. Cortina, and J. Wing. Computational thinking. *ACM SIGCSE Bulletin*, 39(1):195–196, 2007.

[17] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, October 2007.

[18] L. Lessig. *Code And Other Laws of Cyberpace, Version 2.0*. Basic Books, 2006. Also accessible online via a Creative Commons License, http://codev2.cc.

[19] H. Lewis and H. Abelson. Qr48 and csci e-2. Acessed August 29, 2008, http://www.eecs.harvard.edu/qr48, 2007.

[20] E. Nakashima. In child porn case, a digital dilemma. *The Washington Post*, January 16 2008. accessed via http://tinyurl.com/2fnsb4.

[21] J. Niedermeier. In re boucher. http://www.eff.org/files/Boucher.pdf, 2008.

[22] B. Obama. Barack Obama discusses bubblesort at Google. Accessed August 29, 2008, http://www.youtube.com/watch?v=k4RRi_ntQc8, 2008.

[23] C. Papadimitriou. MythematiCS: In Praise of Storytelling in the Teaching of Computer Science and Math. *SIGCSE BULLETIN*, 35(4):7–9, 2003.

[24] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[25] J. Saltzer, D. Reed, and D. Clark. End-To-End Arguments in System Design. *ACM Transactions on Computer Systems*, 2(4):277–288, 1984. Originally published in 1981 in the second Conference on Distributed Computing Systems.

[26] M. Shank. Teacher storytelling: A means for creating and learning within a collaborative space. *Teaching and Teacher Education*, 22(6):711–721, 2006.

[27] J. Sidak. A CONSUMER-WELFARE APPROACH TO NETWORK NEUTRALITY REGULATION OF THE INTERNET. *Journal of Competition Law and Economics*, 2(3):349, 2006.

[28] M. Snir. Computing and information science and engineering: Quo vadimus. Technical Report 13-2008-1, University of Illinois at Urbana Champaign, 2008. http://www.cs.uiuc.edu/homes/snir/PDF/ComputingandInformationScienceandEngineeringTR.pdf.

[29] J. Stewart. John hodgman on net neutrality. Accessed on August 29, 2008, http://www.comedycentral.com/videos/index.jhtml?videoId=114648, July 19 2006.

[30] A. Wang. The Shazam music recognition service. *Communications of the ACM*, 49(8):44–48, 2006.

[31] M. Webster. Merriam-webster online dictionary. website, 2008. referenced on August 29, 2008.

[32] J. Wing. Computational thinking. *Communications of the ACM*, 15(5):33–35, March 2006.

[33] T. Wu. Network neutrality, broadband discrimination. *Journal of Telecommunications and High Technology Law*, 2, 2003.

[34] J. Zittrain. *The Future of the Internet And How To Stop It*. Yale University Press, 2008. Also accessible via a Creative Commons License at http://futureoftheinternet.org.