Overview and Goals of the Proposed CS 111 Revision

Beginning Fall 2014, we will be implementing a revision of Computer Science 111, *Introduction to Computer Science I*. Key features of the new curriculum include:

- a breadth-first introduction to computer science that supplements an introduction to programming with exposure to a variety of other topics, including low-level computer organization and key theoretical concepts like computability
- the use of Python, which provides a simpler syntax and a collection of easy-to-use libraries that facilitate the creation of interesting and engaging programs
- an introduction to multiple programming paradigms, beginning with a functional approach, and continuing with imperative and object-oriented programming
- separate tracks for students with and without prior experience, both of which prepare students for the second course in the major
- introduction of peer instruction to increase retention and increase active learning

The new curriculum is inspired by and based on the introductory Computer Science course that has been offered at Harvey Mudd College since 2006, entitled “CS for All”. This curriculum has been subsequently adopted by a number of other programs. An overview of the curriculum can be found at [http://www.cs.hmc.edu/csforall](http://www.cs.hmc.edu/csforall).

Why Python?

Python is becoming the language of choice in a wide range of computing settings. It has the remarkable property that it is both intuitive and easy to learn, and also a powerful, mature language that is suitable for challenging tasks in high performance computing, software engineering, and data analysis. For these reasons, many Computer Science departments are moving to adopt Python as the first language of choice, replacing Java.

Python has the benefit of extensive libraries for doing high-performance computing, large-scale data analysis, interactive computing, graphics, and more. It also supports modern programming concepts including object-orientation and functional programming. In this respect it combines the best properties of Matlab (engineering and numerical computing), R/SAS (statistical and data centric computing), Java (software engineering and object orientation), and plotting packages. It also has the benefit of a rapidly growing user base creating an ecosystem of contributed packages, due in part to the fact that (unlike Matlab or SAS) it is free and open source.

Goals

Our proposed revision of CS 111 has several goals, all of which we hope to address via the new curriculum:

1. *To continue to provide a rigorous introduction to Computer Science, while making the subject more accessible to students with a broader range of experience, and sparking*
continuing interest and success in CS. The course’s breadth-first approach does a better job of exposing students to the intellectual richness of computer science than does our current approach. Python’s simpler syntax allows for a greater focus on problem-solving and on the creation of interesting and stimulating programs. The course’s use of a functional approach at the start helps to level the playing field between students with differing amounts of prior experience, and the use of separate tracks challenges the most experienced students and prevents the inexperienced students from being intimidated by their experienced peers. All of these factors may help to entice students who weren’t planning to major in computer science to consider doing so, and they should also provide students with a strong foundation for subsequent success in the major.

2. As part of the larger goal of attracting a broader range of students into the major, we hope in particular to increase the number of women choosing to study computer science. Harvey Mudd has seen its percentage of female CS majors increase from roughly 10% before adopting this course to approximately 45% today [3]. This increase can be attributed to several factors -- including practices adopted by their department that are separate from the course -- but among these factors are the way that the course enables students to better understand what computer science is, and its connection and relevance to engaging, real-world problems [1].

3. To give our majors a strong foundation for CS 112 and other subsequent courses. Python allows for a reduced focus on syntax and an increased focus on problem-solving. Starting the course with functional programming and recursion makes it easier for students to master these challenging and important concepts. The unit on low-level organization gives students an accurate mental model for how their programs work, and thus facilitates their understanding of higher-level concepts, including traditionally challenging topics like the use of references/pointers.

4. To give non-majors who take the course a fuller appreciation of computer science as a discipline, as well as useful skills that they can use in their own fields of study. While we see the benefit of continuing to have one or more courses aimed at students who are not likely to go further in CS, we feel that many non-majors would benefit from the revised course, which will combine useful programming skills with a broader exposure to other aspects of computer science.

Assessment

We plan to measure the impact of the revised course in several ways. We will give an end-of-semester survey to CS 111 students this spring and in subsequent semesters to assess the impact of the course on their perceptions of computer science and on their intentions to go further in CS. We can base this survey on one that has been used at Harvey Mudd [2]. We also plan to devise final-exam questions that can be given under both versions of the course to see
whether and how student performance on those questions changes. Finally, we can compare the performance in CS 112 of students coming from both the old and new versions of CS 111.

We will continue to track enrollments in all our courses, and will break down enrollments by gender to try to determine impact on diversity in the major.

Practical Details

- We will begin by transforming the fall 2014 sections of CS 111 into classes that are comparable to the inexperienced sections of the Harvey Mudd course. Students with AP credit would continue to be placed into CS 112. Then, in the spring of 2015, we will introduce an experienced section of the course. We are considering offering the experienced section at the same time as one of the inexperienced sections, so that students could go between the two sections as needed in the early weeks of the semester.

- We will introduce active-learning techniques in CS 111 that will promote student engagement with and understanding of the material. In particular, we intend to use peer instruction, which has been employed successfully in a number of science classes at BU, as well as in computer science classes elsewhere [4].

- The lower syntactic overhead of Python should allow us to reach a comparable level of depth in programming, despite the addition of other topics that are not currently covered in CS 111. For example, console input and file processing are both substantially easier in Python, and will thus require less time. Other factors that should help in this regard include the different sequencing of the topics, and the greater level of engagement that peer instruction elicits.

Longer Term Directions and Connection to other Curricula

- CS 112: Beginning in the spring of 2015, CS 112 will be modified to accommodate the changes to CS 111. In coordination with CS 112 instructors we will add a short initial “bridge” module to CS 112 to transition students from Python to Java. Since students are exposed to object orientation in Python, some key Java concepts will already be familiar.

- In the future, we see the need to decide between two options: (1) continuing to provide a bridge from Python to Java at the beginning of CS 112, and maintaining the use of Java in that course; or (2) transition CS 112 as well to Python, and allowing students to learn Java on their own or in a later course like CS 320. We propose to organize a committee of CS 112 instructors and other interested faculty to address these questions.
References


CS 111: Introduction to Computer Science 1

Requirements

1. Thirteen problem sets. Each problem set will include one “lab” assignment that will be completed during the lab period with TFs and course assistants present for guidance.
2. A number of quizzes (probably three).
3. Final exam.
4. Attendance at and participation in both lectures and labs.

Grading

1. Attendance and participation: 10%
2. Assignments: 40%
3. Quizzes: 30%
4. Final exam: 25%

The final exam will replace your lowest assignment grade if doing so helps your final grade. The final exam will also replace your lowest quiz grade if doing so helps your final grade.

Text

“CS for All”, online at http://www.cs.hmc.edu/csforall/. This text will be adapted for use at BU.

Syllabus


Assignments: Rock/Paper/Scissors (game playing with randomness),

Week 2. Data types, Strings, and Lists.

Assignments: Simple robot exploration (empty room, maze)

Week 3. Introduction to functions. Developing recursive solutions to problems. Recursive list processing.

Assignments: Working with Sequences and Data. Working with Built-in functions.


Assignments: Hangman, Sleepwalking Student (Random Walk), Recursive rendering of turtle graphics.

Week 5. Randomness, simulation. Representation of strings, binary numbers, and binary images. Logical operations.

Assignments: “Lights out” game (problem decomposition, recursion), Caesar cipher.

Assignments: Converting between decimal and binary; arbitrary bases; binary image compression (run-length encoding).

**Week 7.** Von Neumann Architecture. Implementing functions in assembly language.

Assignments: Constructing a ripple-carry adder; 4-bit multiplication circuit.

**Week 8.** Implementing recursion in assembly language. Introduction to For and While loops, imperative programming, and iteration vs. recursion.

Assignments: Basic assembly language programming. Computing the cube of an integer, arbitrary powers, and Fibonacci numbers.

**Week 9.** Nested loops, program structure, role of function definitions and return statements. Constants, local variables, mutable and immutable data types.

Assignments: Creating Mandelbrot set; Monte Carlo evaluation of Pi.

**Week 10.** Two dimensional lists and arrays. Object-orientation.

Assignments: Game of Life, Matrix Multiplication, Creating Magic Squares

**Week 11.** Continuation of object-oriented programming. Dictionaries and files.


**Week 12.** Continuation of object-oriented programming. 3D Graphics, Animation. Search-based artificial intelligence.

Assignments: 3D graphics. AI for Connect-Four.

**Week 13.** Design principles. Algorithm analysis. Hard problems and easy problems.

Assignments: Creating HTML.

**Week 14.** Impossible problems. Formal models of computation, including Turing machines.
Cognate Comment Request

TO: Name: Mark Crovella
Department: CAS Computer Science

FROM: Name: Michael Sorenson
Department: CAS Biology
Telephone: 617-353-6983 E-mail msoren@bu.edu

Course Number: CS 111 Title: Introduction to Computer Science 1

Our Department would like to request cognate comments on this course. A complete course approval form is attached for your review. If you need further information, please do not hesitate to contact me.

Kindly return the signed original to me by __________ so that I may include your comments when submitting our proposal for review and approval. Please do not send any cognate letters directly to the dean’s office. Thank you.

Comments:

The Biology curriculum committee reviewed the proposed revisions to CS 111.

The committee viewed the proposed revisions as positive developments, particularly the adoption of the Python programming language, and thus supports this proposal.

Please explain fully any objections.

Signature: __________
Date: 23 April 2014

Title: Professor and Department Chair
Subject: Re: Request for cognate approval for changes to CS 111

Date: Wednesday, April 16, 2014 at 9:50:02 AM Eastern Daylight Time

From: Lipman, Barton L <blipman@bu.edu>
To: Crovella, Mark E <crovella@bu.edu>

I have no concerns regarding this change.

Is there a form I should fill out to indicate approval or is a letter (or even this email) sufficient?

Bart

On Apr 15, 2014, at 7:10 PM, Crovella, Mark E <crovella@bu.edu> wrote:

Dear Bart,

I’m writing to let you know that the Computer Science department is proposing to renovate CS 111 (Introduction to Computer Science I). As a department with a significant number of majors who take CS 111, we’d like to let you know about this plan and to request your cognate approval.

I’ve attached a detailed rationale as well as a sample syllabus, but here’s an executive summary. Our current CS 111 focuses on “how to program” and teaches the Java language. Our new course will still emphasize developing strong programming skill (with programming assignments due every week), but will place it in the context of a breadth-first introduction to computer science intended to give students a more balanced view of the field of computer science. Along the way we will be adding peer instruction and other pedagogical improvements.

One aspect of the revision will be the switch to teaching Python instead of Java. Besides being a great first language for learning to program, Python is rapidly becoming the language of choice for many real-world problem domains, including numerical/scientific computing, large-scale data analysis, interactive computing, and graphics. We believe that students who learn Python will find it a useful tool across a wide range of problem types and applications.

Please don’t hesitate to contact me if you have any questions. We would like to send this to the NSCC this semester, so a response by April 25 would be helpful.

Best,
Mark

Mark Crovella
Professor and Chair
Department of Computer Science
Boston University
http://www.cs.bu.edu/~crovella/

<CS-111-Syllabus.pdf><CS111-Revision-Rationale.pdf>
Hi Mark,

Thanks very much for sending this to us. The revision looks very compelling. We have gone with Python for our introductory course at the Masters level too, and have found the decision to be a good one. I think it likely to work very well in 111 too. I will forward this change to some of our faculty who work on the undergraduate curriculum to request their input.

Again thanks and best,
-Ben

On Tue, Apr 15, 2014 at 7:19 PM, Crovella, Mark E <crovella@bu.edu> wrote:

Hi Chris (and Ben and Nachi) -

Here in CS we’ve been working for some time on renovating our curriculum in various ways. One aspect is a major overhaul of CS 111 to make it more attractive to a wider range of students, and to present computer science in a more balanced way for newcomers to the field.

Since SMG sends a lot of students to take CS 111, we'd like to request cognate approval for this revision. Below is the message I'd like to get in the hands of whomever is responsible for advising your undergrads who might take CS 111. Can you forward and let me know who to talk to?

Thanks,
Mark

—

Subject: Request for cognate approval for changes to CS 111

I’m writing to let you know that the Computer Science department is proposing to renovate CS 111 (Introduction to Computer Science I). As a department with a significant number of majors who take CS 111, we’d like to let you know about this plan and to request your cognate approval.

I’ve attached a detailed rationale as well as a sample syllabus, but here’s an executive summary. Our current CS 111 focuses on “how to program” and teaches the Java language. Our new course will still emphasize developing strong programming skill (with programming assignments due every week), but will place it in the context of a breadth-first introduction to computer science intended to give students a more balanced view of the field of computer science. Along the way we will be adding peer instruction and other pedagogical improvements.

One aspect of the revision will be the switch to teaching Python instead of Java. Besides being a great first language for learning to program, Python is rapidly becoming the language of choice for many real-world problem domains, including numerical/scientific computing, large-scale data analysis, interactive computing, and graphics. We believe that students who learn Python will find it a useful tool across a wide range of problem types and applications.

Please don’t hesitate to contact me if you have any questions. We would like to send this to the NSCC this
semester, so a response by April 25 would be helpful.

Best,
Mark

Mark Crovella
Professor and Chair
Department of Computer Science
Boston University
http://www.cs.bu.edu/~crovella/