Background: Consideration of Computational and Algorithmic Thinking for MIT Undergraduates

Professor Krishna Rajagopal

former Chair of the Faculty (2015-17)

March 21, 2018

Recent History

- In April 2016, the Chair of the Faculty (me) and the Dean for Undergraduate Education (Prof. Denny Freeman) charged an ad hoc Working Group on Algorithmic Reasoning and Computational Thinking for MIT Undergraduates
  - Profs. Eric Grimson (Chair), Deepto Chakrabarty, Michael Cuthbert, Peko Hosoi, Caitlin Mueller, Jim Orlin, Troy van Voorhis.
- The Working Group released a final report: "A computational thinking requirement for MIT Undergraduates" (March 2017) in which they outlined findings and possibilities for implementation
- The Chair of the Faculty and the Dean for Undergraduate Education asked the Committee on the Undergraduate Program (CUP) to consider some pending questions and how best to proceed with the working group's recommendations (March 2017)
- Today, the Chair of the Faculty Prof. Susan Silbey has asked me to begin this discussion with some reflections on the charge to the Working Group and their report. The Chair of the CUP will then continue.

Work of the Working Group

- Input sought from all faculty and undergraduates. Twice. (The second time after releasing a draft report.) Members of the WG contacted all department heads and met with faculty from every department, soliciting input re the questions in their charge. Engaged with the CUP, FPC, and the Academic Advisory Council of the UA.
- Working Group met extensively (more than ten times) through spring and summer of 2016. Explored how faculty and students across five schools use computational thinking and algorithmic reasoning; intellectual frameworks employed; to what extent these are already being taught; and whether they believe that MIT should acknowledge computational and algorithmic thinking as an explicit expectation of all our graduates.
- Sustained deliberation among the seven members of the WG as they gathered, considered, and synthesized multiple perspectives. No one of them, and I think no one of us, could have done this alone.
- Draft report released end of September 2016. Many comments received, and incorporated after further deliberation. Final report released March 2017.

Findings from Working Group Report

- Computational thinking should play an explicit role in the formal education of all MIT undergraduates. It provides a distinct type of rigorous thought of important intellectual value; requires and develops important modes of communication; acknowledges transformational impact of computation across many disciplines; creates opportunities for our students and graduates.
- The working group recommends that all undergraduates be required to take at least one subject offering in computation.
- Computational thinking involves more than the skill of computer programming or the ability to use computer tools; it includes fundamental modes of reasoning about the rendering of physical or social systems in a manner that enables computational experiments to complement physical or social ones. Formal exposure is important for all students.
- …proceed with a consideration of mechanisms by which a computation requirement could be instituted for all undergraduate students, while addressing the impact adding an additional degree requirement or substituting a current requirement would have on student load and while addressing the need to connect computational thinking to domain-specific contexts across different intellectual disciplines. (cf Duane's presentation.)

What is Computational Thinking?

- The thought processes involved in formulating a problem and expressing its solution(s) in such a way that an information processor can effectively carry out that solution.
- Solve a problem by: abstracting from detail to come up with a sequence of steps; implement the abstraction efficiently; repeat until done.
- An analytical skill that everyone (not just computer scientists) can use to solve problems, design systems, and understand natural phenomena and human behavior.
- Not just computer literacy or learning how to program. The importance of a grounding in computational thinking, for MIT undergraduates, is broader and more foundational than any specific disciplinary needs for programming skills.
- For most of us, grasping computational thinking does require the use of a computer programming language as a framework within which to explore computational and algorithmic concepts. It is also required to go from formulation to solution, as well as to create or explore complex models of social, physical or biological systems.
- Working group also provides a more fine-grained description of what elements are essential to computational thinking.

Elements of Computational Thinking

- Fundamental constructs and their roles in abstraction.
    - Abstraction of processes: capturing common patterns in an algorithmic description that can be generalized and applied to multiple instances.
    - Abstraction of data: capturing patterns within complex collections of data to structure it so that it can be appropriately/efficiently processed.
    - Decomposition and modularity: reducing a computational task to a sequence of simpler operations, each of which is a separate computational tasks, and reintegrating the results of the sub-operations into a solution for the original problem.
    - Iteration and recursion.
- Elements of design for computer programming. Modular design and the role of abstraction. Creating programs that can be shared, understood, tested, debugged.
- Developing skills in at least one programming language.
- Understanding and extending basic classes of algorithms.
- Modeling our physical, biological and social worlds; assessing merits and limits of models; methods for presenting and understanding results of computational modeling, such as visualization and statistical analysis of uncertainty.
- Possible extensions and disciplinary applications include: applications of computational modeling within many domain-specific contexts; understanding the limits of computers; visualization and non-textual interaction; computational creativity.

Computational Thinking for MIT students

- Students should develop skills and modes of thinking so that they can construct or recognize useful, well-written algorithms, and can implement them, and can use them to model physical, biological or social systems. They should learn fundamental skills, and practice them, building a foundation for subsequent exploration of computation in any of a wide range of disciplines.
- The working group gives five reasons why computational thinking should play a role in the formal education of students in all parts of the Institute, why every MIT undergraduate student should be articulate in computational thinking:
- It is a distinct type of rigorous thinking that is of intellectual value.
- It requires and develops important modes of communication.
- Computers are transformational agents in the 21^{st} century.
- It creates opportunities and access for our students and graduates, as computation permeates more and more disciplines and industries.
- It should be, and should be seen as, an essential part of the background that all MIT graduates bring to their roles as professionals and as broadly educated citizens in a world strongly influenced by science and technology.

Implementation

- Goals of implementing a requirement include improving how computational thinking is taught and learned at MIT, including by those students who would take at least one computation subject without such a requirement. (Currently at least 90% of MIT students.)
- Implementing a requirement would make it possible for MIT to say, to its students and to the world, that every MIT graduate understands computational reasoning, computational approaches to physical, biological and social challenges, and how computational agents are changing so many aspects of modern life.
- Options for implementing a requirement, pros and cons of each, were introduced in the working group report.
- These have been further analyzed, and much discussed, by the CUP starting late in Spring 2017. This will be the subject of Duane's presentation.
- Implementation in a different sense, a sense that is also important and that was a part of the thinking behind charging this working group, is already underway.
- Curriculum development prompted by the ideas and findings of the working group is happening, in various departments, at varying scales.

Working Group Report Coda

- The working group unanimously believes that computational thinking is an essential part of the education of every undergraduate student at the Institute.
- It is an intellectual mode of thought of relevance to many intellectual disciplines.
- It develops important modes of communication.
- These factors are in addition to, and in the long run more important than, pragmatic advantages that computational tools provide to an MIT student.
- The working group recommends that the Institute proceed with a consideration of mechanisms by which a computation requirement could be instituted for all undergraduate students, while addressing the impact on student load and the need to connect computational thinking to domain-specific contexts across different intellectual disciplines.

Potential Computational and Algorithmic Thinking General Institute Requirement (COMP GIR) Options

Professor Duane Boning

Chair, Committee on the Undergraduate Program

March 21, 2018

CUP Considering COMP GIR Pros/Cons

- Need for requirement
- Content of requirement (learning objectives)
- Structure of requirement
  - 6+6 or 12-unit
  - Whether or not Departments could specify a subject
  - What Departments can expect, if they cannot specify
  - Ability to AP or ASE out of requirement
- Student flexibility/room for exploration vs. departmental flexibility
- Governance of possible requirement
  - Ability to adapt to changing student needs

Potential COMP GIR Structure

- 12 units in 6 + 6 structure
  - 6 unit intro from set of approved subjects – specific choice could not be mandated by degree program
  - 6 units follow-on from set of approved subjects – subject could be mandated by degree program. Departments might offer subjects in the modes of COMP applicable to and used within their discipline; others might designate subject(s) from another department, or leave choice open to student. The 6 follow-on units of COMP could be part of a larger subject.
- Governance
  - SOCR-/SHR-like subcommittee with oversight of COMP GIR, approving subjects that meet the learning objectives of the requirement at the intro or follow-on level, and update the COMP requirement as needs, expectations, and technologies evolve
- ASE: Students could ASE the intro 6 units and take their 12 units of COMP from follow-on (or more advanced) COMP subjects

COMP GIR Options for Discussion

These are not yet proposals; other/hybrid options possible. Options are meant to help explore and gather thoughts and feedback on pros/cons.

- Option 1: No COMP GIR is instituted.
- Option 2: The REST Requirement is reduced from two subjects to one, with the removed REST replaced by a COMP GIR.
- Option 3: The Science Core is expanded by one group (to 7 total) with addition of 6+6 unit COMP GIR, and students are expected to successfully complete requirements from 6 of the groups.

Impact of REST to COMP (Option 2) on Degrees †

- Degrees with free REST (would be used by COMP)
  - 3A, 4, 4-B, 5, 11, 12, 14-1, 14-2, 15-1, 15-3, 17, 18, 21*, 22, 24-1, 24-2, CMS, STS
- Degrees that now designate COMP-like REST
  - Swap REST with COMP in degree requirements
  - 1-ENG, 2-A, 2-OE, 9, 16, 16-ENG, 20
- Degrees that now have COMP-like degree requirement
  - Swap Departmental requirement with COMP
  - In some cases adds 6 units to degree but remains within 198 unit limit
  - 2, 3, 3-C, 6-1, 6-2, 6-3, 6-7, 6-14, 15-2, 18-C
- Degrees that would lose a REST and need to add units to degree to continue to designate that subject
  - 7, 7-A, 8-Flex, 8-Focused, 10-C; all remain within 198 unit limit
  - 10, 10-B, 10-Eng; adds 6 units, pushing to 201-204 total units

† degree programs may specify up to three GIRs, typically one or more of LAB and the two RESTs; these GIR units are not counted within the limits of the degree program units.

Option 1: No COMP GIR is instituted.	

Pros	• COMP GIR may not be needed to achieve (at least partial) COMP education for majority of students:
	    o 91% of 2016-2017 graduating class completed some form of a COMP subject by graduation
	    o many students have department requirement and/or use electives
	    o many students now taking more advanced CS subjects: >750 in each of 6.009, 6.036 in AY18
	• Students would continue to have flexibility to choose if or how they study computation, within constraints (if any) imposed by their degree program.
	• Avoids adding further complexity to GIR structure.

Cons	• Does not signal that MIT expects COMP knowledge of all students, as in other GIR subject areas. • Does not provide room within the GIRs for students seeking to take COMP within their degree; since not a GIR, access to COMP subjects may differ among students (e.g., less accessible to students coming in with fewer AP units). • Does not encourage departments to incorporate computation into the curriculum. • Does not address potential unevenness or narrowness in the learning objectives and/or pedagogy of existing COMP-like subjects. • Continues to leave "majoring in 6 or 6-X" as a default/generic/poor signal to those seeking students for internships and/or UROPs requiring COMP skills and knowledge. • Does not leverage or foster interaction of CS faculty with other departments, programs, or subjects.
Notes:	• Does not address a call by some community members to evolve the GIRs, explore other topics for inclusion into the GIRs (e.g., statistics, ethics, etc.), and/or to review the GIRs as a whole. • Other mechanisms besides a GIR could be considered to advance computational thinking preparation for all MIT students: allow for more computation subjects to count towards REST/LAB; encourage students to study computational thinking; facilitate departmental efforts to incorporate computational thinking in their departmental subjects and programs.

Option 2: The REST is reduced from two subjects to one, with the removed REST replaced by a COMP GIR.	
Pros	• All students would take at least 12 units of COMP subject(s). In 6+6 structure, the intro 6 units would provide a base of shared knowledge and experience across all undergraduates; follow-on 6 units would deepen that in a flexible way. • Signals MIT belief that COMP is a necessary area of knowledge and mode of thinking, in order for our students to most effectively interact in a data- and computation-intensive world and to be prepared for, and help shape, the continued evolution of that world. • Enables and encourages departments to develop follow-on COMP subjects that integrate computation with their departmental curriculum (in the 6+6 model). • Enables subjects throughout the curriculum to assume or depend on base COMP knowledge, potentially enriching many subjects across MIT.
Cons	• All students would be required to take COMP subject(s) even if they are not interested in COMP. • Loss of REST reduces flexibility in some majors (that don't specify both RESTs). • Since all students would be required to take COMP subject(s), a variety of subjects accommodating various levels of prior experiences would be needed.
Notes	• Most degree programs as currently structured (according to degree charts) could accommodate this change without major changes to their curriculum: they could trade the REST for the COMP, assuming computation subjects approved are similar to those currently being discussed. ○ 1.00, 1.000, 2.086, 6.00, 6.0001 + 6.0002 are all REST subjects. • Raises question about purpose and effectiveness of remaining single REST. Still needed? • Departments would need to assess content in new COMP-intro subjects as prerequisites for follow-on subjects, and adapt their follow-on COMP and other subjects as needed. • Does not address a call by some community members to evolve the GIRs, explore other topics for inclusion into the GIRs (e.g., statistics, ethics, etc.), and/or to review the GIRs as a whole. • A 6+6 model likely encourages early completion of the first 6 units. Issue of 1st year credit limits and interaction with COMP subjects (now encountered with 6.0001) will remain or potentially increase.

Option 3: The Science Core is expanded by one group (to 7 total) with addition of 6+6 unit COMP GIR, and students are expected to successfully complete requirements from 6 of the groups.	
Pros	• Students would have increased flexibility/choice within the Science Core. • COMP would be a valid early and/or late Science Core option for students, within the 17 GIRs.
Cons	• Science Core subjects would not all be required: lose signal/expectation that all of the current Science Core content is necessary as part of an MIT education. • Departmental programs could not assume all Science Core groups as prerequisites. • Early student selection of Science Core subjects could have implications for future (students may need to take an additional subject for a particular major, even if completed Science Core). • Students would not have a "common experience" with the Science Core. • Some GIR groups/subjects may have small or varying numbers of students. Likely that many students would elect not to take Biology or Chemistry GIR.
Notes	• If MIT believes that all of the skills/modes of thought/knowledge in the individual Science Core GIRs (with computation) are distinct and necessary, then MIT should require all of them. • Existing COMP-like subjects are (mostly) REST subjects now; would they still be counted as RESTs in the 6 of 7 approach? • Possible variant: Leave Science Core at current 6 subjects; instead students could pick 3 of 4 from set of 2 REST, LAB, COMP requirements. Degrees would still only specify 3 GIRs. • Departments would need to assess content in new COMP-intro subjects as prerequisites for follow-on subjects, and adapt their follow-on COMP and other subjects as needed. • Does not address a call by some community members to evolve the GIRs, explore other topics for inclusion into the GIRs (e.g., statistics, ethics, etc.), and/or to review the GIRs as a whole. • A 6+6 model encourages early completion of the first 6 units. Issue of 1st year credit limits and interaction with COMP subjects (now encountered with 6.0001) will remain or potentially increase.

Open Discussion

These are not yet proposals; other/hybrid options possible. Options are meant to help explore and gather thoughts and feedback on pros/cons.

- Option 1: No COMP GIR is instituted.
- Option 2: The REST Requirement is reduced from two subjects to one, with the removed REST replaced by a COMP GIR.
- Option 3: The Science Core is expanded by one group (to 7 total) with addition of 6+6 unit COMP GIR, and students are expected to successfully complete requirements from 6 of the groups.