1. Please describe and provide a rationale for the proposed change to the existing degree:

The College of Arts & Sciences Department of Computer Science (CS) proposes to modify requirements for the BA in Computer Science by adjusting the balance between specifically required and elective courses in the total of 15 (four-credit) CS courses required for the major. Currently, 11 of those 15 courses are specifically required and 4 are unrestricted electives. In the proposed distribution, 5 “foundation” courses are specifically required; 2 “formal tools” and 2 “central topics” courses are structured electives, in each case chosen from a menu of three; and 6 courses are unrestricted CS electives (as detailed on pp. 3-4 below). This proposal does not entail the addition of new courses or the deletion of current offerings; in fact, a student who so chooses will still be able to take the full set of 11 currently required courses toward his/her total of 15 (as though nothing had changed). For other students, however, increased flexibility will make it possible to focus a rigorous program of study on advanced topics of individual interest within the CS curriculum.

There are a number of reasons for making this change, but the fundamental driver is that the field of Computer Science is growing. In the past decade, quite a few subfields in computer science have matured significantly, and the impact of Computer Science across academic disciplines as well as across society has diversified. The net result is twofold: compared to a decade ago, there are a wider range of topics within Computer Science that excite and attract students, and there is a wider set of motivations and goals that students have when entering the CS major.

While the Department has been discussing these trends for some time, our recent Academic Program Review (completed in Spring 2013) helped to crystallize both an appreciation of these trends and a commitment to address them in revising our curriculum. Of the five specific actions that the Department proposed in response to our APR, four are directly or indirectly concerned with our teaching mission, and two specifically target curricular renovation.
In January 2013, the Department began an extensive effort to redesign our curriculum. The CS Department Chair, then Stan Sclaroff, formed a committee of six faculty from a range of sub-disciplines in the department: Systems (Rich West), Theory (Peter Gacs), Networks (Mark Crovella, also committee Chair), Cyber Security and Cryptography (Leo Reyzin), Applications (Wayne Snyder, who is also our Director of Undergraduate Studies), and our Senior Lecturer (Dave Sullivan). All CS faculty members were notified of the upcoming peer study and formation of the undergrad curriculum committee via email on 1/21/13. In response to a strong expression of interest in participating in the committee, another theorist (Leonid Levin) was added to the committee following the initial peer study.

The committee led a four-step process. Each of these steps included open discussion via email. First, the committee conducted an in-depth peer study, reviewing all aspects of the CS major requirements at 31 top or peer institutions. This culminated in a report to the faculty and discussion at the department faculty meeting on 2/27/13. Next, the committee surveyed each faculty member in hour-long interviews on their views of what changes to our curriculum were desirable. This culminated in a report to the faculty and discussion at the department faculty meeting on 4/10/13. Open discussion of options then took place, leading to another faculty meeting on 5/2/13. The current proposal was formulated and opened for discussion at that meeting. Since the semester was at an end, a vote by email was scheduled for July 1, 2013. Voting was understood to be public. Although some faculty sent their votes to the committee chair directly, all individual faculty votes were announced at the end of voting. In the end, 15 faculty members voted in favor of this proposal, with two opposed and two abstaining.

One of the principal goals of this effort, flowing from our APR recommendations, was to improve our engagement with, and education of, students who may have diverse interests and may come into the major with varied goals and motivations. To accomplish this, the faculty decided to adopt a notion of “tracks,” i.e. sets of courses that each allow a student to study deeply in one particular area of Computer Science. The intent is to appeal to students who may be attracted to Computer Science because of one of its subfields — for example, data science, cyber security, or game development — and to provide them with a set of courses that give them a solid and satisfying foundation in their chosen area. In fact, our peer review had identified tracks as an innovation that many of our peers had undertaken for this same reason.

Our decision to broaden our curriculum by adding tracks is one element of an overall strengthening of our curriculum. According to the final report of our external visiting committee: “The computer science dept. at Boston University has fallen behind...its narrow undergraduate curriculum attracts only 1.5% of the students; only 15% of the student body is taking computer science classes....” The committee goes on to recommend (inter alia): “Create a choice of tracks to meet the diverse interest of the study body...The department is encouraged to study recent curriculum updates undertaken by top universities for reference.”
In keeping with our goals, it will *not* be a requirement that students complete a track in order to satisfy the major requirements. We fully expect that many students will be interested in studying broadly or eclectically, or may not have specific areas of interest as CS majors. We do not want to restrict the options of such students. In other words, we do not believe that the deeper focus enabled by tracks is a necessary part of becoming a computer scientist, but rather that for some students it is a very desirable part.

Our faculty frankly struggled with how to offer a meaningful program of tracks in a major that requires students to take only 4 elective courses. A track that consists of 3-4 courses in one subfield would exhaust all or nearly all of a student’s choices. The faculty considered and debated a wide range of options and alternatives, balancing the addition of more choice against the need to provide students with a strong foundation as delivered in our required course set. In the end, we decided that an increase in electives was needed to make a system of tracks viable.

To add choices, the faculty broke the major down into four groups of courses. The reason for the breakdown was to ensure that, given a wider set of options, students still receive essential instruction across specific sets of topics. These groups are as follows:

Group A: Foundational Courses. These are five courses that are essential parts of the major and that all students must take.

Group B: Formal Tools. These are courses that provide training in basic algorithms and formal methods. Students must take at least two (of three) from this group.

Group C: Central Topics. These courses are more advanced than Groups A and B, and cover broadly important subfields of CS. Students must take at least two (of three) from this group.

Group D: Electives. These are advanced courses that provide in-depth study of specific topics. Students must take 15 courses in total, so depending on how many Group B and C courses they take, they will take between four and six courses from Group D.

This structure changes the degree requirements as follows: Under the old system, all courses in Groups B and C were required. In the new system, the student is required to take 2 of the 3 courses in each group.

The effect of this change is to give students who want it the opportunity to take a larger set of advanced courses (from Group D) in fulfilling the major. Hence, it creates a more demanding course of study for those students, who will develop competency in a larger set of advanced topics. These courses are often more project-oriented than courses in Groups B and C, and hence more rigorous in terms of applied skills. This will help to correct a deficiency of our program, as commonly noted in our graduating students’ exit surveys as well as the report of our external review committee.
All students who complete the CS major will have completed at least three courses that are principally formal (math-oriented): CS 131, and two from Group B. Further, our students are all expected to have had Calculus (MA 123 or equivalent). For comparison, the following schools specifically require four or fewer such math or mathematically oriented courses: Berkeley (BA), Brown, Dartmouth, Duke, Caltech, Chicago, Columbia, Cornell, Michigan NYU, Princeton, Rochester, Texas, and Wisconsin.

One element of our degree that has not changed is the fact that when a Math course is listed as an alternate prerequisite for a CS course, the math course counts in place of the CS course toward the major requirements. Currently, the faculty has determined that the following substitutions are allowable: MA 242 for CS 132, MA 294 for CS 235, and MA 381 for CS 237.

Finally, it is important to reiterate that, for students whose intellectual orientation is strongly formal/ mathematical, it is still possible to complete the major exactly as before, taking all the courses in both Groups B and C, and only four advanced electives. Advising will continue to play an appropriately key role in ensuring that all students match their interests and goals to best possible pathways through the major.

2. **Please describe how the proposed change(s) advances the Strategic Plan of the school/college and of the University plan:**

Commitment 3 of the CAS Strategic Plan sets out the principles by which the College will provide outstanding undergraduate education. Our proposal seeks to raise the standard of education in Computer Science and, as such, is strongly in support of that commitment.

With specific respect to “Recruiting great Boston University students,” the renovated Computer Science major is a more interesting, varied, and modern program of study than before. We expect that strong prospective students will compare it favorably with programs elsewhere and so, be more likely to apply and to attend BU. With respect to “retaining and ensuring the success of CAS students,” the proposed modifications to the major will make it easier for students who discover CS late in their time at BU to complete the major without summer work or extra semesters. This is consistent with the desire to support a liberal arts and sciences education for CAS students.

At the University level, a distinct commitment has been made to computer and mathematical sciences in the form of the new building planned for Commonwealth Avenue. Our proposal seeks to modernize the CS curriculum at a time when the University and College are investing in the program. The University has made innovative STEM education a priority, including a commitment to diversity among STEM students, which our proposal is designed to enable.

3. **Please list the program requirements for the current and revised programs: (expand the table as needed and denote new courses in bold print)** The proposal involves no new courses.
Old Program:

The Major in Computer Science requires a total of 15 courses, all completed with a grade of C or higher, and distributed as follows:

CS Background Courses (7)

CS students are expected to be comfortable with standard high school mathematics as well as calculus at a level equivalent to completion of CAS MA 123.

Background courses are those listed as prerequisites or co-requisites in the course descriptions of CS Required Courses, currently:

- CS 111 Introduction to Computer Science I
- CS 112 Introduction to Computer Science II
- CS 131 Combinatoric Structures
- CS 132 Geometrical Algorithms
- CS 210 Computer Systems
- CS 235 Algebraic Algorithms
- CS 237 Probability in Computing

The list of prerequisites or co-requisites to CS required courses may contain a CS course “A” with a non-CS alternative “B.” In such a case, course “B” may be counted instead of course “A” as a CS background course. Refer to the course descriptions of Required Courses for details.

CS Required Courses (4)

- CS 320 Concepts of Programming Languages
- CS 330 Introduction to the Analysis of Algorithms
- CS 332 Elements of the Theory of Computation
- CS 350 Fundamentals of Computing Systems

CS Elective Courses (4)

- Four CAS CS courses numbered 400 or higher

In special cases, a student, with consent of the instructor, may petition the Undergraduate Director to count course other than those listed in the Courses section. Attention is called to the following: CS courses number 600 and above; 500-level courses in Mathematics & Statistics; and Computer Engineering course offered by the College of Engineering.

CS major credit will not be given for CS 211, CS 212, courses below CS 111, or for Metropolitan College courses.
New Program:

The Major in Computer Science requires a total of 15 courses, all completed with a grade of C or higher.

CS students are expected to be comfortable with standard high school mathematics as well as calculus at a level equivalent to completion of CAS MA 123.

The 15 courses required for the major are divided into four groups: A, B, C, and D. Students must take all courses in Group A, at least 2 courses from Group B, at least 2 courses from Group C, and a total of 15 courses from Groups A-D.

Group A: Foundational Courses
These courses build essential foundations for the study of CS. Students must take all five.
- CS 111 Introduction to Computer Science I
- CS 112 Introduction to Computer Science II
- CS 131 Combinatorial Structures
- CS 210 Computer Systems
- CS 330 Introduction to the Analysis of Algorithms

Group B: Formal Tools
These courses provide training in basic algorithms and formal methods. Students must take at least two from this group.
- CS 132 Geometrical Algorithms
- CS 235 Algebraic Algorithms
- CS 237 Probability in Computing

Group C: Central Topics
These courses are more advanced than Groups A and B, and cover broadly important subfields of CS. Students must take at least two from this group.
- CS 320 Concepts of Programming Languages
- CS 332 Elements in the Theory of Computation
- CS 350 Fundamentals of Computing Systems

Group D: Electives
- All CS courses at the 400- and 500-levels

The list of prerequisites or co-requisites for a CS course may contain a CS course "X" with a non-CS alternative "Y." In such a case, course "Y" may be counted toward the major instead of course "X."

In special cases, a student, with consent of the instructor, may petition the Undergraduate Director to count course other than those listed in the Courses section. Attention is called to the following: CS courses number 600 and above; 500-level courses in Mathematics & Statistics; and Computer Engineering course offered by the College of Engineering.

CS major credit will not be given for Metropolitan College courses.
4. Please list learning outcomes for the revised program:

Students graduating with a major in Computer Science are expected to:

1. Understand and evaluate the organization, design, and construction of hardware and software systems for computing.

2. Attain a level of mathematical ability allowing the student to formally abstract and analyze computational processes.

3. Analyze problems that require computation, and design and implement appropriate solutions that are efficient and effective.

These are unchanged from our current program.

5. How does the change place your program in the context of programs at peer institutions?

As mentioned above, the faculty reviewed requirements for the CS Major at 31 institutions: Berkeley, Brown, Cal Tech, Chicago, CMU, Columbia, Cornell, Dartmouth, Duke, GWU, Georgia Tech, Harvard, Johns Hopkins, Maryland MIT, Michigan, NYU, Northeastern, Penn, Princeton, Rochester, Stanford, Tufts, UCLA, UCSD, UIUC, UMass-Amherst, USC, UT-Austin, Washington, and Wisconsin. Of these, 9 offer tracks (by some name): Columbia, Cornell, Georgia Tech, Rochester, Stanford, GWU, U Mass-Amherst, Michigan, and Washington. Thus, our adoption of tracks will enable us to be comparable and competitive with these top schools.

More generally, we found that variation and choice are the norm among other CS programs. Almost every program we looked at provides options for students to tailor the major to their interests.

The current proposal can be described as reducing the number of “non-elective” (meaning, specifically required) courses in the major. Our existing program includes 11 non-elective courses, and the new program includes 9 non-elective courses (without decreasing the total number of courses required to complete the major). Of the 31 institutions surveyed above, only 5 required as many non-elective courses as in our existing program. The most common number of non-elective courses across surveyed institutions is in the range 6-8.

6. How does the change affect other academic units?

No specific effects. Computer Science collaborates closely with the neighboring Department of Mathematics & Statistics. In some cases, as currently, students majoring in Computer Science will be allowed to substitute MA for required CS courses. A cognate memo of endorsement is attached from the chair of Mathematics & Statistics, Professor Tasso Kaper.
7. How will you notify current students of the proposed changes and implement the requirements?

2014/2015 Bulletin, CS web site, CS advising, CAS Advising and Orientation for incoming students

8. Please document any implications that the change has on professional accreditation or licensure at the program or school/college level:

N/A

9. Please list the resources needed including IT, new faculty, new staff, reassignment of faculty from existing courses to new ones (especially if the existing course(s) is not being removed from the bulletin), technology enhanced classrooms, office space, and other facilities:

No new resources are specifically required by this program change. There will undoubtedly be some shift in demand among courses as a result of this change. We will continue to keep careful track of patterns and adapt as we go.