ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #32009

X+CS: A Computing Pathway for Non-Computer Science Majors

Dr. Susan Mitchell, University of Maryland, Baltimore County

Dr. Susan Mitchell is a Senior Lecturer at the University of Maryland, Baltimore County (UMBC). She holds a PhD in Information Systems from UMBC and a MS in Computer Science from The Johns Hopkins University. She has been a teaching faculty member in computer science for 27 years, with research in the areas of software process, software process improvement, knowledge management, and computer science education.

Dr. Katharine Cole, University of Maryland Baltimore County

Dr. Katharine Cole has served as the UMBC Vice Provost and Dean of Undergraduate Academic Affairs since 2017. Dr.Cole earned a Bachelor's degree in Biological Sciences from the University of Delaware, a Master's degree in Zoology from the University of Northern Colorado and a Ph.D. in Pathology from the University of Maryland School of Medicine. Dr. Cole has served as an Assistant Dean of Graduate and Undergraduate Studies in the College of Arts andSciences at the University of South Florida, as a university-level Associate Dean for Undergraduate Studies at the University of South Florida, and as Associate Provost at the University of Tampa. In her administrative roles she has overseen the general education curriculum, first year student programs, Bachelor degree program curricula, academic policy,Center for Teaching and Learning, Academic Success Centers as well as other student success offices and initiatives. In these roles Dr. Cole has worked collaboratively with deans and the faculty to developing high impact practices designed to provide undergraduate students with equal opportunities to gain knowledge and skills by integrating multiple disciplines in a meaningful way. Prior to her administrative roles, Dr. Cole's research interests were in the broad area of the anti-neoplastic mechanisms of novel chemotherapeutic compounds, specifically targeting the role of inositol hexaphosphate (IP6) in cell cycle regulation and cellular differentiation in malignant and benign human neoplastic cells. She has published twenty-two peer-reviewed papers, holds two patents related to replicative human liver cell line, has been the principle investigator/co-PI on five grants from non-profit organizations and has mentored 4 master/doctoral students. During her time in faculty positions at the University of Maryland School of Medicine and the University of South Florida, she taught nine different courses in the areas of human physiology, human pathology and biological imaging. Dr. Cole's professional development as a scientist and an educator has provided her with an innovative perspective and an ability to critically analyze meaningful data as well as the organizational aptitude necessary to successfully accomplish new university strategic academic initiatives and develop strategic academic programs that benefit the university community.

Prof. Anupam Joshi, UMBC

©American Society for Engineering Education, 2020

X+CS: A Computing Pathway for Non-Computer Science Majors

Susan M. Mitchell, Anupam Joshi, Katharine E. Cole

University of Maryland, Baltimore County

Abstract

With computing impacting most every professional field, it has become essential to provide pathways for students other than those majoring in computer science to acquire computing knowledge and skills. Virtually all employers and graduate and professional schools seek these skills in their employees or students, regardless of discipline. Academia currently leans towards approaches such as double majors or combined majors between computer science and other non-CS disciplines, commonly referred to as "CS+X" programs. These programs tend to require rigorous courses gleaned from the institutions' courses for computer science majors. Thus, they may not meet the needs of majors in disciplines such as the social and biological sciences, humanities, and others.

The University of Maryland, Baltimore County (UMBC) is taking an approach more suitably termed "X+CS" to fulfill the computing needs of non-CS majors. As part of a National Science Foundation (NSF) grant, we are developing a "computing" minor specifically to meet their needs. To date, we have piloted the first two of the minor's approximately six courses. The first is a variation on the existing Computer Science I course required for majors but restricted to non-majors. Both versions of the course use the Python language and cover the same programming content, but with the non-majors assigned projects with relevance to non-CS disciplines. We use the same student assessment measures of homework, projects, and examinations for both courses. After four semesters, results show that non-CS majors perform comparably to majors. Students also express increased interest in computing and satisfaction with being part of a non-CS major cohort.

The second course was piloted in fall 2019. It is a new course intended to enhance and hone programming skills and introduce topics such as web scraping, HTML and CSS, web application development, data formats, and database use. Students again express increased interest in computing and were already beginning to apply the computing skills that they were learning to their non-CS courses.

As a welcome side effect, we experienced a significant increase in the number of women and under-represented minorities (URMs) in these two courses when compared with CS-major-specific courses. Overall, women comprised 52% of the population, with URMs following a similar upward trend.

We are currently developing the third course in the computing minor and exploring options for the remaining three. Possibilities include electives from our Information Systems major. We will also be working with our science, social science, and humanities departments to utilize existing

courses in those disciplines that apply computing. The student response that we have received thus far provides us with evidence that our computing minor will be popular among UMBC's non-CS population, providing them with a more suitable and positive computing education than existing CS+X efforts.

1. Background

With computing impacting most every professional field, it has become essential to provide pathways for students other than those majoring in computer science to acquire computing knowledge and skills. Virtually all employers and graduate and professional schools seek these skills in their employees or students, regardless of discipline. However, students who are non-computer science (CS) majors attempting to take CS classes run into the fact that, outside of introductory programming, most CS classes are designed to be taken by majors. These classes make assumptions about the math and science background of students and are aligned to cover concepts in a certain sequence and depth appropriate for CS majors. CS classes also cover topics that are required for students going to graduate school in CS or working for tech companies such as Google and Microsoft, but that are not very useful for students from other majors that will use CS in their disciplines. Examples include disk or memory management, IP protocol details, computational complexity theory, Turing machines, etc. The current design of most CS classes thus misses out on a more diverse set of students interested in computing.

Academia currently leans towards approaches such as double majors or combined majors between computer science and other non-CS disciplines, commonly referred to as "CS+X" programs, to provide diversified computing opportunities [1]. However, these programs tend to require rigorous courses gleaned from the institutions' courses for their computer science majors, thus likely being unsuitable to non-CS fields. They also tend mostly to combine computer science with mathematics and natural science disciplines such as physics and biology. Consequently, they may not meet the needs of majors in disciplines such as the social sciences and humanities. They also have not had uniform success. For instance, Stanford discontinued its CS+X pilot last year [2]. Many universities also offer a minor in CS, but it again requires students to have the math and science background to take CS courses. For example, at UMBC, students wanting a CS minor need Calculus I and Discrete Math.

Other existing approaches to providing computing education for non-CS majors include concentrations in CS, such as the University of British Columbia's "mini-streams" [3] and the University of Washington's computer science course options for non-majors [4]. However, methods such as these do not lead to an academic attainment such as a certificate or minor, possibly lacking attraction and proper guidance for non-CS majors.

2. "X+CS" at UMBC

The University of Maryland, Baltimore County (UMBC) is a mid-size public research university. It is a minority-majority institution with approximately 11,000 full-time undergraduate students. There are approximately 1,400 undergraduate computer science majors.

UMBC is taking an approach that we term "X+CS" to fulfill the computing needs of non-CS majors by developing a "computing" minor specifically for these students. The effort is part of a National Science Foundation (NSF) EAGER (EArly-concept Grants for Exploratory Research) grant. EAGER grants are used ". . . to support exploratory work in its early stages on untested, but potentially transformative, research ideas or approaches. This work could be considered especially 'high risk-high payoff' in the sense that it, involves radically different approaches, applies new expertise, or engages novel disciplinary or interdisciplinary perspectives." [5] UMBC's computing minor will be different from a traditional CS minor. The minor will include a set of basic required courses common to all who desire a computing background and a flexible set of electives germane to students' particular majors. To date, we have piloted the first two of the computing minor's approximately six courses with promising results. We are also shepherding the proposed minor through our internal approval processes.

We see two kinds of students that our computing minor approach will serve. The first are students who will major in their field of interest but will supplement their major with the computing minor. The second are students who will define new interdisciplinary majors rooted in computing using UMBC's Individualized Study (INDS) program in which students follow a core curriculum and a set of approved courses from a variety of disciplines. These two groups cover a large number of students interested in computing but not wanting to be computer science majors.

3. Piloting of the First Computing Minor Course

The first course in UMBC's planned computing minor is a variation on the existing Computer Science I course required for majors but restricted to non-majors. It is a four-credit course that includes a hands-on laboratory component. Both versions of the course use the Python language and cover the same programming content, but with the non-majors assigned homework and projects relevant to non-CS disciplines. Examples include statistical analysis, working with large data sets, and data visualization. No programming experience is required for either version. The programming techniques covered include modularity, abstraction, top-down design, documentation, debugging, and testing. Core topics include control structures, functions, lists, strings, abstract data types, file I/O, and recursion.

Our first offering was in the spring of 2018 with eighteen students. We restricted enrollment to majors in other than computer science, computer engineering, physics, chemistry, engineering, and mathematics. The goal was to be able to make a comparison between students not required or less likely to enroll in a computer science course and computer science majors. The course was offered as a "special section" of Computer Science I. The prerequisite for Computer Science I is Precalculus. As many social science majors do not follow the same mathematics path as computer science and natural and physical sciences majors, non-majors were admitted with either Precalculus or Applied Calculus. A comparison of UMBC's Precalculus and Applied Calculus shows a comparable level of mathematical skill.

We used the same student assessment measures of homework, projects, and examinations for both the majors and non-majors sections. (There were three majors sections with a total

enrollment of approximately 320 students.) For this first semester, students were even given the same exams (a midterm and a final). Results show that the non-majors section performed overall as well as the sections for majors.

The section has been offered three more times (fall 2018 through fall 2019). Further outcomes, student feedback, and lessons learned are provided in the Discussion section below.

4. Piloting of the Second Computing Minor Course

The second computing minor course was piloted in fall 2019 and taught by the first author. It is a new three-credit lecture course intended to enhance and hone programming skills and introduce practical topics relevant to non-majors. The prerequisite is Computer Science I, either a CS-majors or non-majors section. As this was a pilot, the course was offered as a "special topics" course entitled Continued Computing for Non-Computer Science Majors. Enrollment was minimal (seven students) mainly due to the course counting towards no university or major requirement and difficulty in spreading the word, especially mid-year, about the offering. However, we will continue to offer the course in order to gather more data regarding its effectiveness and appeal to non-CS majors.

For the pilot, the basic course design was to continue the development of students' programming skills for the first half of the semester, then introduce general computing skills and technologies. The programming portion of the course was taught in Python using both the Linux and PyCharm development environments. Topics included extended Python syntax, classes and objects, using pre-defined Python libraries, creating one's own Python libraries, version control, data formats, and using a Python integrated development environment for program development, testing, and debugging. Students mainly programmed using their own computers, but had access to UMBC's Linux system, which they had used in Computer Science I.

The second half of the course introduced topics such as HTML and CSS, web application development, web scraping, data visualization, and database use. Technologies introduced included Bootstrap (for HTML and CSS), BeautifulSoup (for web scraping), Flask (for web application development), and gnuplot (for data visualization). These technologies were chosen because of their popularity, free availability, and their ability to be installed on a variety of operating systems. It was emphasized to students that the end goal of using these technologies is not to become an expert in any of them, but to learn the subject areas that they support.

Assessment for this pilot was by projects only, assigned every other week. We wished to make the course as concept-driven and hands-on as possible, given the unique opportunity of such a small enrollment. Students were overall satisfied with the format and content of the course and performed well. Further outcomes, student feedback, and lessons learned are provided in the Discussion section below.

5. Discussion

5.1 First Course

We have now offered the non-majors section of Computer Science I for four semesters (spring 2018 through fall 2019). All were taught by the same instructor (the first author). In the fall of 2019, we opened enrollment to physics, chemistry, engineering, and mathematics majors to increase enrollment. Composite results for all four semesters show that non-CS majors receive a similar total number of As, Bs, and Cs as CS majors (Table 1). The DFW rate (the total number of D's, F's, and withdrawals) are also comparable.

	All Non-CS Majors Sections		**All CS Majors Sections, Fall 2019**	
	Quantity	**Percent**	**Quantity**	**Percent**
A	22	23	186	31
B	31	32	170	29
C	16	16	78	13
D	5	5	42	7
F	6	6	45	8
W	17	18	69	12
Total:	97		590	
DFW Rate:		29%		26%

Table 1. Comparison of Majors vs. Non-majors Final Grades

Students also express satisfaction with being part of a non-CS major cohort and increased interest in computing. When asked in a fall 2019 end-of-semester survey if students thought that it was better to have been in a Computer Science I section specifically for non-majors, the following were typical sentiments.

> I do think it was better because the sections for majors contain programming experts, whereas the people in the non-majors section are less advanced.

> I think it was better for me to be in the CMSC 201 non-majors section because I was able to work on projects and homework assignments I had a greater interest in.

When asked about an increased interest in computing, a common reaction was that students were pleased that they now see how coding can help them in their field of study and that programming had been demystified.

> Yes, I want to learn more about what I can do in coding and how to apply it to my field. It is challenging sometimes but it can also be enjoyable.
>
> Yes. Coding used to seem like a very foreign concept that I never really would have been able to grasp, but now I feel confident in my ability to possibly continue onward with it.

Although interest in computing generally increased, most students responded that they would not be continuing with computing courses. The main reasons given were that they did not have room in their schedule the following semester, that the class was very time consuming, or that they would be graduating. This raises two points. The first is that students need a firm incentive to take further (or even a first) computing classes. At UMBC, Computer Science I does not satisfy a general university requirement and satisfies a major requirement for very few majors outside of computer science and computer engineering. Second, the majority of students who take the non-majors section are upperclassmen (see Table 2). We speculate that offering a computing program that students would enter at an early stage in their academic careers and also receive a minor degree would significantly encourage incorporating computing into their undergraduate education. In the meantime, we need to formulate better methods to reach and motivate underclassmen.

Semester	Freshman	Sophomore	Junior	Senior	Other
Spring 2018	1	2	6	8	1
Fall 2018	7	3	5	6	2
Spring 2019	4	5	6	7	0
Fall 2019*	3	7	8	9	0
Total:	15	17	25	30	3
Percent:	17	19	28	33	3

*Does not include seven students who withdrew

Table 2. Enrollment by Year (from rosters)

Several valuable and enlightening observations were made over the course of the four semesters that we have offered the first course. For example, a first-day student survey given each of the four semesters reveals that non-majors in general have very little to no programming experience. They typically use computers to create documents, spreadsheets, and presentations, and for shopping, email, gaming, and social media. Very few have performed any significant amount of programming. This is in sharp contrast to entry-level computer science majors. Although this may be expected, it underscores the point that instructors must be aware of this difference in backgrounds and provide appropriate pedagogical support for non-majors.

A second observation is that non-majors are not especially comfortable with the general use of computers. Each semester, the instructor held a one-class “install-fest” to help students install Python, a professional text editor, plotting software, and file transfer software on their laptops.

Although, with assistance, students are able to install the software, many have difficulty understanding the difference between programming on their own computers and using client software to program using the remote UMBC Linux system. They have an especially hard time navigating their computer's file storage system and distinguishing between files that are stored locally on their computer and those stored remotely.

A last observation is that although non-majors overwhelmingly own portable computers, they are not necessarily suitable for programming. Many are old and slow and some students simply own tablets. This is obviously a problem in general and especially for students who have little to no programming experience.

As a welcome side effect to the offering of the Computer Science I non-majors section, we experienced a significant increase in the number of women when compared with CS-major-specific sections. Over the four semesters offered, women comprised 54% of the population (97 students), as compared with 26% in the majors sections for the fall 2019 semester (590 students). URMs followed a similar upward trend, with 30% black/African American and Hispanic students, as compared with 24% for the fall 2019 semester. Details are shown in Table 3 and Table 4, respectively.

Semester	Enrollment	Male	Female	Other	Prefer Not to Answer
Spring 2018	18	7	11	0	0
Fall 2018	23	14	9	0	0
Spring 2019	22	11	11	0	0
Fall 2019	27*	11	16	0	0
Total:	90	43	47	0	0
Percent:		48	52	0	0

*Does not include seven students who withdrew

Table 3. Enrollment by Gender (self-identified)

Semester	White/ Caucasian	Black or African American	Hispanic or Latino	Asian/ Pacific Islander	American Indian or Alaskan Native	Multiple Ethnicity	Other	Prefer Not to Answer
Spring 2018	10	6	1	1	0	0	0	0
Fall 2018	4	5	1	10	0	1	1	1
Spring 2019	8	7	0	4	1	1	1	0
Fall 2019*	10	7	0	9	0	1	0	0
Total:	32	25	2	24	1	3	2	1
Percent:	36	28	2	27	1	3	2	1

*Does not include seven students who withdrew (total of 90 students)

Table 4. Enrollment by Ethnicity (self-identified)

5.2 Second Course

The second course took some time to reach a good stride. The course format began as strictly lecture. However, as time passed, it was clear that students were not engaged. Midway through the semester, we changed the format to lecture and demonstration during the first class of the week and hands-on, in-class exercises during the second. End-of-semester surveys showed a preference for the more interactive, hands-on approach.

Students again expressed increased interest in computing and were already beginning to apply the computing skills that they were learning to their non-CS courses. The following are representative end-of-semester survey comments.

> Yes! I used what we learned in web-scraping in my BIOL 302L lab. I needed to get data about different viruses from a website, but there were over 200 viruses and many fields for each virus. I wrote a web-scraper to go through the viruses I was interested in and collect information about the characteristics that I specified; I was able to produce a CSV in order to analyze my data.
>
> I have maintained my website on the UMBC GL server. This isn't specifically for a class but it is academic related. Also this class has made me more comfortable scripting in MatLab which I have used in MATH 404 (partial differential equations).

Among the lessons learned from this pilot is that students struggle more with the programming aspects of the course than with the more practical topics presented in the second half of the semester. Given that the minor is a "computing" minor, perhaps more time should be given to programming or possibly the practical topics should be moved to a third course altogether.

Students also expressed the most interest in the course's web-related topics, such as web scraping and web application development. It would be interesting to explore whether this is due to the applicability of these topics, if they are simply more fun, or if there is another reason.

6. Summary and Future Plans

Our offerings of the first two courses in UMBC's planned computing minor have shown success both in terms of student performance and overall satisfaction. The first course, which has been taught four times, is currently offered as a special section, with variation, of our Computer Science I for CS majors. Students report satisfaction with being in a section with only non-majors and an increased interest in computing, both in general and relative to their particular majors. As a welcome side effect, we have experienced a significant increase in the women and under-represented minority populations in comparison to students who typically enroll in Computer Science I sections for CS majors.

We are continuing to offer the non-majors section of Computer Science I with the goal of integrating it into our planned computing minor. We have not made the decision whether to keep it as a special section of Computer Science I or to convert it to its own course. An advantage to keeping it as a special section is that students may use it as part of a computer science major if they choose to change to computer science or pursue a computer science minor. A disadvantage is that the course would remain closely tied to the majors sections' common syllabus and programming language.

Students in our first offering of Continued Computing for Non-Computer Science Majors, a special topics version of the second course in our planned computing minor, also express satisfaction. Additionally, they report applying the skills that they learn to other courses and personal pursuits while they learn them. We are continuing to offer the course as a special topics course and will convert it to a regular course with a new title when appropriate. As the one section that has been offered had very low enrollment, we need to gather more data as to student performance and satisfaction and course content.

The student response that we have received thus far provides us with evidence that our computing minor will be popular among UMBC's non-CS population, providing them with a more suitable and positive computing education than existing CS+X efforts. There is also evidence that our X+CS approach will provide an opportunity to increase the participation of women and URMs in computing.

Two concerns that we need to address as we are developing the computing minor are difficulty in gathering enrollment for the first two courses and the need to catch students early in their academic careers. Enrollment in the first course is gradually increasing by word of mouth and through direct marketing to non-CS majors. But it is below what is desired. The majority of students tend to be majors in the biological sciences, some of which require that their students take either a computer science or mathematics elective. We need to determine ways to reach out to and motivate other non-CS majors early in their academic careers, especially in the social sciences and humanities.

We are currently developing the third required course in the computing minor. It is a combination data structures and algorithms course oriented towards practitioners. This is different than the data structures and algorithms courses generally required for computer science majors. For practitioners, it is important to know what data structures exist, what they are useful for, under what circumstances particular data structures are used, and how to use standard libraries that implement them. Similarly, it is useful for practitioners to have a high-level sense of algorithmic strategies and the notion of time complexity, without necessarily needing to know algorithmic proofs and theoretical details.

We are also in the process of determining suitable electives for the remaining three courses for the computing minor. Possibilities include courses from our Information Systems major and the development of new non-majors courses within our Computer Science and Electrical Engineering Department. We will also be working with our science, social science, and humanities departments to utilize existing courses in those disciplines that apply computing.

7. References

[1] C. Ruff , “Computer Science, Meet Humanities: in New Majors, Opposites Attract,” *Chronicle of Higher Education*, vol. 62, no. 21, January 28, 2016. [Online]. Available: https://www.chronicle.com/article/Computer-Science-Meet/235075. [Accessed February 27, 2020].

[2] J. Leighton, “CS+X Pilot to be Discontinued End of Spring Quarter,” Stanford Report, January 24, 2019. [Online]. https://news.stanford.edu/2019/01/24/csx-pilot-discontinued-end-spring-quarter. [Accessed February 27, 2020].

[3] https://www.cs.ubc.ca/students/undergrad/mini-streams

[4] https://www.cs.washington.edu/academics/ugrad/courses/non-majors

[5] https://www.nsf.gov/about/transformative_research/submit.jsp