Teaching Computer Science at UVA

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Bennett Ross

Spring, 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Yonghwi Kwon, Department of Computer Science

Teaching Computer Science at UVA

CS4991 Capstone Report, 2021

Bennett Ross
Computer Science
University of Virginia
School of Engineering and Applied Science
Charlottesville, VA, USA
lr4nw@virginia.edu

ABSTRACT

My summer internship at Capital One where I was tasked to design a web-based document content search API prompted me to consider the ways that the Computer Science program at the University of Virginia (UVA) should be restructured to better prepare students for experiences similar to mine and produce higher quality software engineers. As a graduating computer science student at UVA who participated in the 2019-2020 new curriculum, I have had the opportunity to experience the early stages of the curriculum that will be adopted in the next academic year. My track in the CS department includes courses such as Data Structures and Algorithms (DSA), Software Development Essentials (SDE), and Computer Organization and Architecture (COA) which is now called Computer Systems and Organization. These courses gave me a much stronger foundation of computer science principles than my peers in the traditional curriculum. Though the future curriculum will do a better job than its predecessor of preparing CS students for employment, it too can be improved. Students will benefit from acquiring more knowledge about the most common algorithms, more experience building applications in the language used in most web development positions (Javascript), and more exposure to cloud computing services.

1. Introduction

The one question that every computer science professor asks is “What is the best way to prepare my students for the workforce?” The University of Virginia’s Computer Science department has asked this question as well; and as a result, they have given their students the option to share an internship experience and provide details on how their tenure as a student has prepared them for the task. As a fourth-year computer science student at the University of Virginia who participated in the 2019-2020 new curriculum, I have a unique perspective of being able to compare my internship experience and knowledge with peers who were either in the curriculum or on the traditional path of the Bachelor of Science in Computer Science (BSCS) track. During my most recent internship at Capital One, I was given the task of designing a document content search web API using Javascript and leveraging several Amazon Web Services (AWS) platforms such as load balancers, simple queue services, lambda functions, databases, and elastic search in order to maximize efficiency. Throughout my experience, I had to become extremely familiar with Javascript, learn all of the different AWS platforms that were utilized, participate in an agile work environment, and craft presentations for both technical and non-technical coworkers.

2. Related Works

In this section, I present related work on learning strategies for computer science education, in addition to the course syllabi and learning objectives for the computer science department at UVA.

In their technical paper, Tychonievich and Sherriff explain the reasoning behind the computer science curriculum overhaul as low overall curriculum cohesion along with being very outdated (2021). The low cohesion was caused by slight and individual tweaks to specific classes that gradually decreased the cohesion of the curriculum. This change also addresses a multitude of inefficiencies due to the length of time that the curriculum has gone without redesign. For example, there has been a tenfold increase in student enrollment and an updated pedagogy to stay current with the latest fields in computing since the last redesign. The paper also provides more detailed insight into their process of redesigning the curriculum. The course syllabi of the 2019-2020 new curriculum courses can be found at http://pilot.cs.virginia.edu/category/courses

The Engineering is Elementary (EiE) program by the Museum of Science in Boston, Massachusetts has developed a culmination of 7 research-based classroom strategies for teaching computer science (Team, 2021). Among the list includes creating real-world relevance and providing opportunities for collaboration and sharing. At the college level creating real-world relevance would be the equivalent of producing software that a company would potentially have you design.

These sources influence the recommendation of changing the Software Development Essentials project. Altering the project will provide the opportunity to create real-world relevance and get students familiar with the working environment. Utilizing local cloud computing platforms will help keep students up to date on a growing topic in the field which was another aim for the redesign.

3. Curriculum Comparisons

The traditional BS Computer Science curriculum requirements include eight classes. The program starts with the introductory programming course taught in Python, then moves to Java and the essentials of software development in the next course. Then the focus switches onto data structures and representation while using C++ and instructional programming languages created by UVA, in addition to taking digital logic design which teaches logic concepts. Then students are required to take Computer Architecture, Operating Systems, and Advanced Software Development which all teach important aspects of software development and how a computer works.

In addition to restructuring the required courses, program coordinators also restructured how and when the material is taught. The material in the traditional track is still taught in the new curriculum, but the sequencing makes the material much easier to digest and to apply to both prior and future knowledge.

The pilot program restructures the courses by requiring seven courses all with two levels:

- Discrete Mathematics and Theory (DMT) which replaces Discrete Mathematics and Theory of Computation;
    - The former course of Discrete Mathematics focused on teaching the principles of logic through the use of functional programming (specifically in Lean); its replacement will divert the attention to proving theorems and writing prose proofs by hand. This will result in students learning the foundation of discrete mathematics and logic instead of pattern matching while programming in Lean. The specifics of how it will cover the Theory of Computation aspects are not posted, but it is safe to assume that the course will be almost identical to its current format.
- Data Structures and Algorithms (DSA) which replaces parts of Data Structures and Representation and Algorithms;
    - Data Structures and Representation covers several topics, but what DSA takes over is the implementation of basic and advanced data structures along with evaluating asymptotic time and space complexity. It also changes the main language from C++ to Java. The second part of the course cover the concepts learned in Algorithms through the use of programming assignments where the algorithms must be implemented and written assignments to write proofs.
- Computer Systems and Organization (COS) which will replace Digital Logic Design, Computer Architecture, Operating Systems, and parts of Data Structures and Representation;
    - The first part of COS will cover the remaining material of Data Structures and Algorithms, Digital Logic Design, and Computer Architecture. It teaches programming from the most basic machine level to high level, ensuring students understand the basic execution model and assembly. It also has lessons on logic gates covered in Digital Logic Design along with teaching all of the concepts covered in Computer Architecture. The second level of the course will cover the more expansive knowledge of Operating Systems; the assignments covered in both courses are almost identical.
- Software Development Essentials (SDE) which replaces Advanced Software Development.
    - Both courses are designed to teach students all necessary software design principles to efficiently implement a complex software entity that involves many aspects of modern software systems. They differ in the language used to implement this entity, SDE uses Java and its Swing framework while Advanced Software utilizes Python and its Django framework.

4. Results

My track on the revised BSCS curriculum was much more efficient than the traditional track. I have found that my knowledge retention of earlier topics taught throughout the curriculum is much higher than that of my peers who came in with the same technical background but did not choose the pilot program. The relatively early exposure to algorithms, within my second semester, as opposed to my fourth or later, was also extremely beneficial to me when applying to internships or even practicing technical coding challenges on LeetCode.

My education has even allowed me to help peers understand concepts and code on assignments from courses on the traditional track that I had not yet taken. Computer Architecture and Operating Systems are commonly regarded as the hardest computer science courses offered; however, I was able to effectively TA for both while only taking Operating Systems the semester prior and never taking Computer Architecture.

5. Recommendations

While the new curriculum is much better than the traditional, it still did not fully prepare me for my internship experience. For example, I had no prior experience with coding in Javascript, presenting my codebase to both technical and non-technical peers, and working with agile software development. If the Computer Science curriculum was restructured just slightly more, future students could at least be exposed to these aspects of computer work. Not only would they experience less of a shock in their internship experience, but they would also be likely to receive internship offers before the summer after their third year, which would give them an even larger advantage in the workforce.

UVA should continue to infuse more algorithms earlier in the CS degree track in addition to teaching Javascript and local cloud computing structures and programs to expose students to popular fields and languages. These changes would be extremely beneficial to UVA students looking to enter the workforce since most applications include some sort of front-end service. Also, cloud computing is an ever-growing field worth over $200 billion right now with the expectation to triple by 2027 (Fortune Business Insights, 2021).

Algorithms are a crucial concept in computer science. An algorithm refers to a "set of steps used in completing a certain task or to get [the] desired output" (How, 2021). They are applied to data structures, or an "orderly arrangement of data" (How, 2021) to optimize the performance of completing a task such as searching, addition, deletion, or sorting; and they are used to find the "best possible way of solving a problem" (Mulong, 2021). Algorithms and data structures are interrelated and complementary. The way data structures are taught with the new computer science curriculum in Data Structures and Algorithms 1 is very effective and needs little improvement because spending about a week on every data structure is enough to gain an in-depth understanding. However, even though we covered the different data structures and their respective methods in one semester, we did not make the connection between how each data structure's methods related to the common practice algorithms within Computer Science.

Cloud Computing is a term that refers to an area of information technology that touches the following areas: "hardware infrastructure, data center facilities, virtualization technologies, and software engineering concepts" (Bellasio, 2020). Most companies in the world utilize the cloud for managing their infrastructure (Bellasio, 2020). Currently, the UVA student's only avenue to learning cloud computing is when taking the introductory course CS4740: Cloud Computing, which is a well-designed course; but only having one course to reinforce the ideas of cloud computing is not sufficient for a field that is growing so rapidly.

A common misconception is that cloud computing is too advanced to be introduced to students earlier than it currently is. At present students take the pre-requisites and then hope to gain admission into the course, but they do not need any coding ability to start learning the cloud (Bellasio, 2020). Public and private cloud computing services that can help non-software developers learn about cloud computing include AWS, Azure, and Google Cloud. If someone without any experience can learn it, these students who intend to major in CS could be introduced to it during their second semester.

My suggestion is that the project normally assigned in Software Development Essentials could be changed into one that includes a cloud-like service such as serverless which can mimic using the cloud without the price of the cloud. Another idea for changing the project is to use Javascript instead of Java. The reasoning behind this is that many applications have some type of user interface so students could learn libraries such as React that are widely used throughout the industry and only offered through Javascript.

6. Conclusion

The upcoming change to be adopted by the CS curriculum in the 2021-2022 academic year is a step in the right direction towards preparing students to enter the workforce. The new approach provides a stable foundation of computational knowledge that cannot be said about the traditional path. But it is not without limitations. Students would benefit from gaining exposure to algorithms at the start of their education so those concepts can stay at the forefront of their minds. In addition to that, switching their SDE application language to Javascript and incorporating elements of cloud computing would provide them with more applicable real-world programming experience, setting them apart from other potential applicants in the workforce.

7. Future Work

If adopted, these changes will give UVA a better foothold for teaching CS and sending skilled workers into the job market. The next phases would include gathering data from students who experience these changes and journey into the workforce. It may also be beneficial to develop new course plans that incorporate the ideas suggested in a logical way for the educational development of students.

References

Bellasio, S. (2020, January 23). Learn Cloud Computing: Prerequisites. Retrieved from https://cloudacademy.com/blog/prerequisites-to-learn-cloud-computing-introduction/

Fortune Business Insights. (2021, August 11). Cloud Computing Market to Hit USD 791.48 Billion by 2028; Rising Demand for Improved Virtual Access to Information among Industries to Foster Steady Growth: Fortune Business Insights™. Retrieved from https://www.globenewswire.com/news-release/2021/08/11/2278451/0/en/Cloud-Computing-Market-to-Hit-USD-791-48-Billion-by-2028-Rising-Demand-for-Improved-Virtual-Access-to-Information-among-Industries-to-Foster-Steady-Growth-Fortune-Business-Insights.html

How Data Structures and Algorithms are important for Computer Science Graduates - Synergisticit. (2021, April 12). Retrieved from https://www.synergisticit.com/how-data-structures-and-algorithms-are-important-for-computer-science-graduates/

McConnell, J. J. (1996). Active learning and its use in computer - JMU. cs.jmu.edu. Retrieved from https://users.cs.jmu.edu/adamses/Web/CS430/Lectures/p52-mcconnell.pdf

Mulongo, C. (2021, August 05). The Importance of Algorithms in Computer Programming. Retrieved from https://www.technotification.com/2019/02/importance-of-algorithms-programming.html

Sherriff, M. (2019). *Courses*. UVA CS 2020 Curriculum Pilot. Retrieved from http://pilot.cs.virginia.edu/category/courses

Team, E. E. (2021, May 17). *7 research-based classroom strategies for Teaching Computer Science (CS)*. EiE Blog. Retrieved from https://blog.eie.org/7-research-based-classroom-strategies-for-teaching-computer-science-cs