

Crafting a Foundation for Computing Majors

David P. Harvie
United States Military Academy
West Point, New York
david.harvie@usma.edu

Tanya T. Estes
United States Military Academy
West Point, New York
tanya.estes@usma.edu

Michael J. Kranch
United States Military Academy
West Point, New York
michael.kranch@usma.edu

ABSTRACT

This paper describes and evaluates a sophomore level survey course in the computing disciplines of computer science and information technology. This course is novel among ABET accredited computer science and information technology programs in the breadth of topics covered and that it serves as a common foundation to both computing disciplines. In addition, students are introduced to advanced computing topics that they may later choose to pursue further in upper-level electives. This paper discusses the motivation of a course for both programs and concludes with the results, challenges, and opportunities for future iterations. This single computing survey course helps students to ensure they selected the correct major early in their academic career. Additionally, it introduces advanced computing topics that students may choose later to pursue in electives.

KEYWORDS

Survey Course, Foundational Course

ACM Reference Format:

David P. Harvie, Tanya T. Estes, and Michael J. Kranch. 2018. Crafting a Foundation for Computing Majors. In *The 19th Annual Conference on Information Technology Education (SIGITE '18)*, October 3–6, 2018, Fort Lauderdale, FL, USA. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3241815.3241857>

1 INTRODUCTION

The purpose of this paper is to describe and assess the efficacy of a sophomore level survey course in the computing disciplines of computer science and information technology. This paper will describe the motivations, objectives, and constraints that factored into the construction of the course, and then it will assess whether the objectives were achieved with a focus on how the course should be improved for future iterations.

During the academic year of 2016–17, the curriculum the United States Military Academy (USMA) was revised in order to allow more disciplinary depth and student choice. One of these changes included having students select their major in the Spring of their freshman year rather than Fall of sophomore year, as had been done in the past. This earlier selection of major meant that students had less knowledge about the programs of study available to them going into that critical decision. Also, students at USMA have the constraint that they must finish their undergraduate degree in four

years. The authors wanted to give the students choosing from one of our computing disciplines, specifically computer science or information technology, the most flexibility to change majors without penalty.

2 MOTIVATION

The computing faculty found that novice computing students often did not fully understand the difference between the computer science and information technology programs. The combination of an introductory programming course and survey course of computer science and information technology topics for all computing majors at the beginning of their sophomore year will either reinforce their decision or encourage them to switch majors without penalty. Additionally, exposure to diverse upper-level computing topics will foster student interest in these subjects which they can pursue in greater depth by taking the appropriate elective(s).

Finally, the course was designed to inspire the computing students by providing a better understanding of the culmination of these two academic programs. Faculty had observed that novice students, especially those with no prior computer science or information technology background, became easily discouraged by some of the of the tougher initial courses. This course would provide the students a glimpse of what they could do in these majors and give them motivation to persist during the challenging courses.

3 COURSE DESIGN

This course, designated CY355 Cyber Foundations – Computing, had to achieve USMA's Academic Program Goal Five of Science, Technology, Engineering, and Math. Specifically, students must achieve the objective of "Explain and apply computing and information technology concepts and practices in the context of the cyber domain" [10]. This objective must be met by all graduates, and, since a student could potentially change to a non-computing major, the proposed course had to ensure that this objective was met.

The initial basis for designing CY355 was to modify the existing intermediate-level information technology course required of nontechnical majors. The existing course covered topics such as databases, web applications, networking, and cyber security [9]. The proposed course would also address these topics but in a more rigorous computing focused way. For example, nontechnical students in the existing course would learn Microsoft Access and complete a simple database project. However, computing students in the new course would learn relational databases using MySQL, which requires them to understand both the database concepts and the specific SQL syntax, and then demonstrate those concepts in both a database project and on examinations of SQL code. In addition, students in the proposed course would learn NoSQL database through

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor, or affiliate of the United States government. As such, the United States government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for government purposes only.

SIGITE '18, October 3–6, 2018, Fort Lauderdale, FL, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5954-2/18/10...\$15.00

<https://doi.org/10.1145/3241815.3241857>

MongoDB. In the existing course, nontechnical students would use Microsoft Expression Web to connect pre-built forms to an existing Microsoft Access Database. On the other hand, computing students would learn JavaScript and build a dynamic website that required the integration of HTML, CSS, MongoDB, and JavaScript using the Meteor framework with no pre-built forms or wizards.

Another facet of this course was the introduction of advanced computing topics that interested students could choose to pursue as upper-level elective courses. The course introduced each advanced topic with an individual lesson interspersed throughout the semester. The advanced topics covered included machine learning, human-computer interaction, encryption, and ethical & legal considerations of security and privacy.

The topics covered and the number of lessons per topic are shown in Table 1. Several of these subjects such as networking, databases, security, and ethics appear in both SIGITE and SIGCSE Top 10 Current and Recommended topics [2].

Table 1: CY355 Topics Covered

Topic	Lessons
Intellectual Property	1
Digitization	6
Exponential Growth	1
Relational Database	5
NoSQL Database	2
Introduction to Machine Learning	1
Human-Computer Interaction	1
Web Site Development	6
Web Application Development	5
Networking Basics	4
Cyber Security	3
Introduction to Encryption	1
Ethical & Legal Considerations	1

Since CY355 is a foundational course to both CS and IT majors, it needs to address topics in both the Computer Science Curricula 2013 [3] and Information Technology Curricula 2017 [6]. Core Tier 1 topics are topics that should be required for every CS program. A CS program should cover 80% of the Core Tier 2 topics with a goal of 90-100% coverage. Table 2 shows the Core Tier 1 and 2 topics that are covered or introduced in CY355. Essential Domains are those topics that all students in an IT program must achieve. Table 3 shows the IT Essential Domains that are covered or introduced in CY355.

4 BENCHMARKING

The Computer Science (CS) and Information Technology (IT) majors at USMA are ABET accredited. Students in both computing majors must take the introductory programming course and the Cyber Foundations – Computing course during the Fall of sophomore year. In order to benchmark this approach, the authors used the ABET Find an Accredited Program search engine [1] to find accredited undergraduate CS and IT programs using “Computer Science” and “Information Technology” as the key search terms. The search engine returned 310 institutions offering an accredited

Table 2: CS Core Tier 1 & Tier 2 Topics Covered [3]

Graphics and Visualization/Fundamental Concepts
Human-Computer Interactions/Foundations
Information Assurance and Security/Foundational Concepts
Information Assurance and Security/Threats & Attacks
Information Assurance and Security/Cryptography
Information Management/Database Systems
Networking and Communication/Introduction
Networking and Communication/Networked Applications
Social Issues and Professional Practice/Intellectual Property
Social Issues and Professional Practice/Civil Liberties

Table 3: IT Essential Domains Covered [6]

Cybersecurity Principles/Cryptography Overview
Information Management/Database Query Languages
Networking/Foundations of Networking
User Experience Design/Perspectives and Impact
Global Professional Practice/Ethical, Legal, and Privacy Issues
Global Professional Practice/Intellectual Property
Web and Mobile Systems/Technology

Bachelors Degree in CS and 37 institutions offering an accredited Bachelors Degree in IT. Comparing the two sets of institutions, there were 28, including USMA, that offered both accredited CS and IT undergraduate degrees.

Reviewing the program requirements for the other 27 institutions, 15 of these schools had courses common to both the CS and IT majors. Upon further inspection of the 15, the most popular common CS and IT courses were programming (7), data structures (4), and introduction to computer science (4). The course proposed by the authors is novel in the breadth of topics covered as a foundational course for both CS and IT majors. However, the amount of programming topics introduced makes this a more complicated introductory course.

5 COURSE EXECUTION

The inaugural course was offered in Fall 2016 with an enrollment of 70 students of which 67 were either CS or IT majors taking the course concurrently with the programming fundamentals course during their sophomore year. The students were instructed in the following languages in order to learn the topics in Table 1:

- MySQL [7] (relational database)
- MongoDB [5] (NoSQL database & web application)
- HTML/CSS/JavaScript (web sites & web application)
- Meteor [4] (web application)

Student comprehension of these languages was primarily assessed using out-of-class projects. Students’ ability to write and understand SQL was assessed using a database project. A programmer’s portfolio project that started with web design which was transformed into a static web site which ultimately became in a web application design assessed student comprehension of HTML, CSS, JavaScript, and Meteor (a JavaScript framework). In-class graded

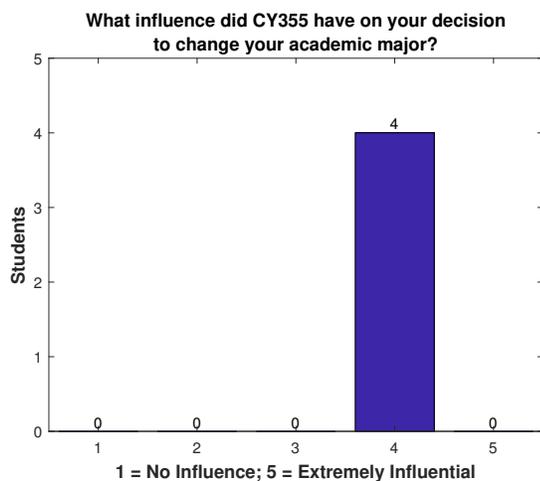


Figure 1: Influence of CY355 to Change Major

exams were also employed to test the student’s comprehension of these languages.

The number of languages introduced during this semester-long course was a challenge to both instructors and students. To mitigate this impediment, the authors added a commercially available online training, CodeSchool.com [8], to the curriculum. The online content was assigned as homework, and then in-class exercises and out-of-class projects built upon the foundation of that online content.

6 RESULTS

The instructors observed that students most easily grasped MySQL, HTML, and CSS, which enabled the rapid writing of small pieces of code with immediate feedback. MongoDB was more challenging as students had to learn the Binary JSON (BSON) format. Additionally, the fundamental concepts and application of a non-relational database is more difficult than a relational database.

By far the most difficult topic was Meteor which is a JavaScript framework to build full stack web applications. There was no existing online content in Code School for Meteor. The students struggled with the integration of HTML, CSS, JavaScript, and Blaze (a Meteor front end language). The students had to troubleshoot multiple layers of code which is difficult. Another obstacle to learning Meteor is that the framework relies not only on JavaScript but also runtime environments such as NodeJS which can be daunting to beginning computing students.

Several weeks after the conclusion of the course, the authors surveyed the former students in order to receive feedback regarding the course. Twenty-nine students (43.2%) of the 67 sophomores responded to the survey. Twenty-eight of these respondents were CS majors at the beginning of CY355, and one was an IT major. Four of the respondents changed their major during the semester in which they took CY355. As shown in Figure 1, CY355 had a large influence on those four respondents. One of these students commented, “It made me realize that working with networks and the general concept of managing a network interested me more.”

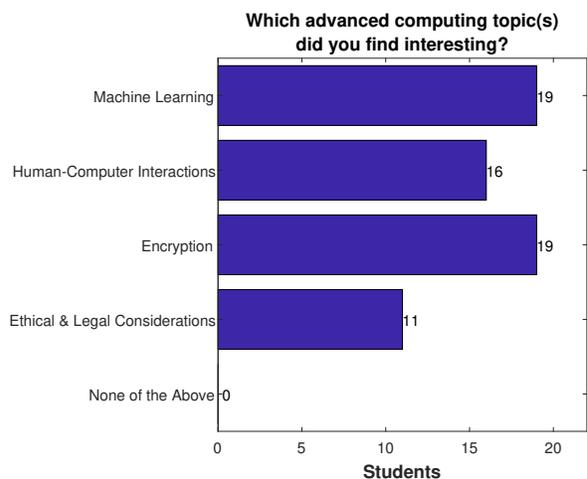


Figure 2: Interest in Advanced Computing Topics

Five of the 29 respondents indicated that they had changed one or more computing electives after taking CY355. Four of those five respondents indicated that CY355 was influential on their decision, and the remaining respondent indicated that the course had minimal impact on their decision. The advanced computing topics covered in CY355 (Machine Learning, Human-Computer Interactions, Encryption, and Ethical & Legal Considerations of Privacy & Security) did generate interest among all the respondents with Machine Learning and Encryption being the most popular among the four topics as shown in Figure 2. A majority of the respondents recommended keeping the advanced computing topics in CY355 as shown in Figure 3.

Students that recommended keeping the advanced topics in CY355 generally stated they enjoyed learning about future possible research areas. One of these students stated, “I actually enjoyed them. They provided a sneak peak into topics which I had never really talked about or considered.” The students that did not recommend keeping the advanced topics generally stated that the introduction of these subjects, even at a cursory level, was not appropriate for an introductory computing course. One of these students stated, “Without prior background it is really hard for students to understand an advanced topic.”

7 CHALLENGES WITH LANGUAGES

As anticipated, one of the greatest difficulties with CY355 was the breadth of languages introduced to the students. Students in the computing majors simultaneously learned Python in their introductory programming course. The respondents stated confidence in their ability to write both HTML/CSS as shown in Figure 4 and in SQL as shown in Figure 5. These results are logical because the students had a dedicated SQL database project to reinforce learning and HTML/CSS is not too difficult to comprehend. The students had less confidence in their ability to write MongoDB code as shown in Figure 6. Comprehension of MongoDB was primarily assessed during the Web Application project in conjunction with Meteor.

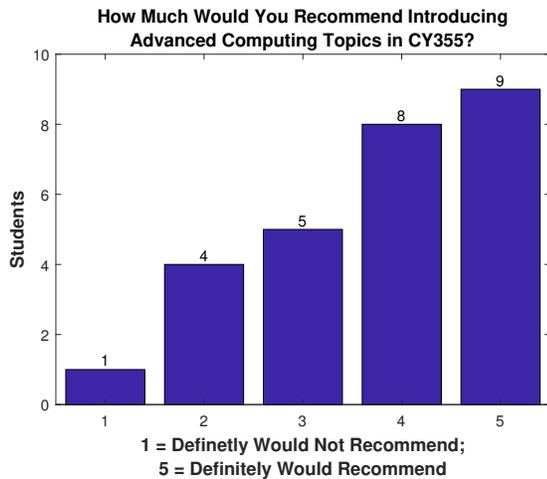


Figure 3: Recommendation to Keep Advanced Computing Topics

However, there was not a separate assessment that focused solely on MongoDB.

As shown in Figure 7, most respondents have little to no confidence in their ability to write Meteor code. The Meteor JavaScript framework can be complex for a novice computing undergraduate student. The Meteor website has an online tutorial where a learner can build a Meteor to-do list. Nonetheless, the blogging and commenting features required in the portfolio were much more complex. As a result, completing the Meteor to-do list exercise did not directly translate in the students being able to write their own Meteor code. In retrospect, the transition from a static web site using HTML/CSS to a dynamic web application using Meteor may have been too abrupt as executed during this inaugural offering of CY355.

When queried on their recommendation to introduce such a broad range of programming languages in CY355, the respondents were almost evenly split in whether to keep or reduce the number of programming languages as shown in Figure 8. The students who appreciated being introduced to different languages recommended that this aspect of the course be maintained. However, most of these same students cautioned how difficult JavaScript and Meteor were to learn. One student stated, “I think that introducing a broad range of languages is a good idea. However, going into JavaScript was probably taking it a little bit too far.”

8 CONCLUSION & FUTURE RESEARCH

Overall, CY355 successfully achieved the two goals that motivated the development of this course. First, CY355, in conjunction with the introductory programming course, confirmed that students chose the correct computing discipline or informed their decision to change majors. Four of the respondents indicated that they changed their major during the semester, and all four indicated that CY355 played a key part in that decision. A good follow-up question would have been what major did they subsequently select. Unfortunately,

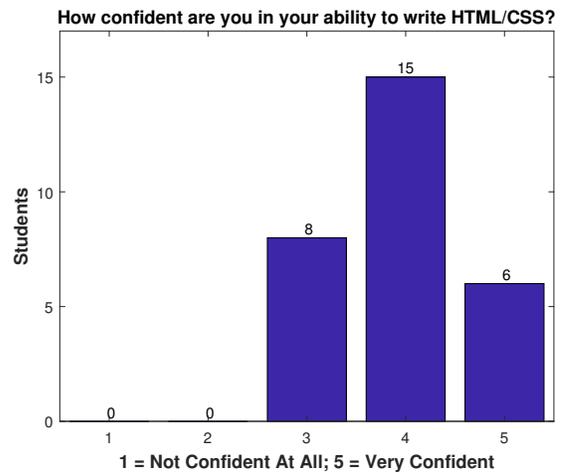


Figure 4: Confidence in Writing HTML/CSS

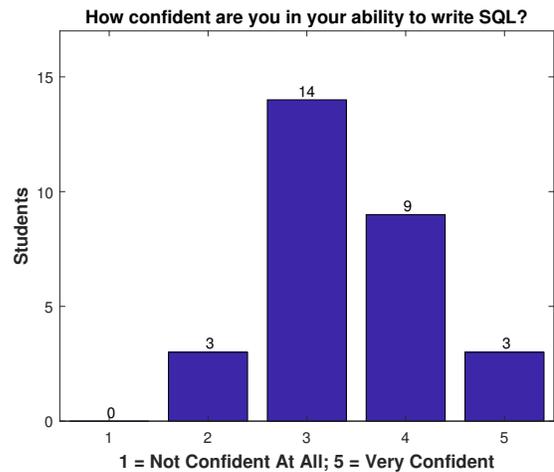


Figure 5: Confidence in Writing SQL

that question could not be asked without potentially identifying the respondent.

Second, this survey course achieved success in introducing students to advanced computing topics they may wish to pursue in greater depth later in their academic career. It is significant that interest was shown in all four advanced topics and that no respondent indicated, “None of the Above”, when asked, “Which advanced computing topic(s) did you find interesting?” There was also a clear majority that favored retaining the introduction of the advanced subjects.

In spite of the positive feedback, there is clearly room to improve the course. The broad range of programming languages is a challenge to both teach and learn. The online training did facilitate some of the learning, but there was no existing online training for the most difficult component, writing Meteor code. In the future iteration of CY355, there needs to be a smoother and more logical transition from static HTML/CSS to the dynamic JavaScript

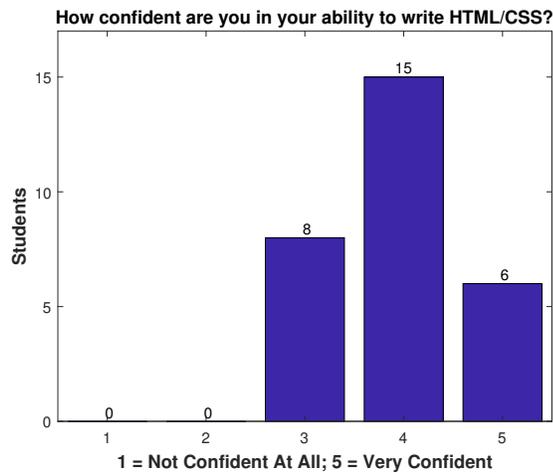


Figure 6: Confidence in Writing MongoDB

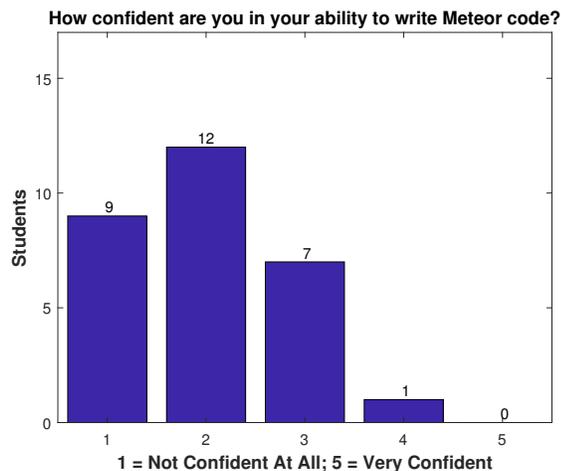


Figure 7: Confidence in Writing Meteor Code

using Meteor. This may be done by building the static website with HTML/CSS/JavaScript that reads in a static file with the intent of then using that model to build the dynamic web application. Another challenge not specifically addressed in this paper is the need to successfully download and install the necessary programming environments on student machines. A future iteration of CY355 may look at the use of Virtual Machines to eliminate the requirement for students to install various programming tools, environments, and libraries on their personal computers.

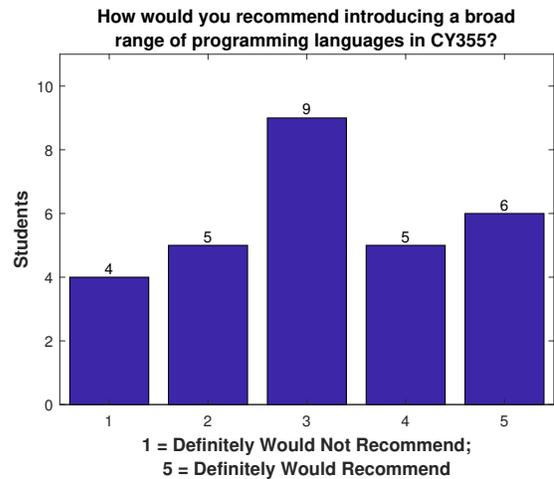


Figure 8: Recommendation to Keep Range of Languages

REFERENCES

- [1] ABET. 2018. Find an ABET-Accredited Program. Retrieved May 29, 2018 from <http://main.abet.org/aps/Accreditedprogramsearch.aspx>
- [2] Joseph V. Elarde and Fatt-Fei Chong. 2011. Introductory Computing Course Content: Educator and Student Perspectives. In *Proceedings of the 2011 Conference on Information Technology Education (SIGITE '11)*. ACM, New York, NY, USA, 55–60. <https://doi.org/10.1145/2047594.2047610>
- [3] Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, New York, NY, USA. 999133.
- [4] Meteor. 2018. Meteor. Retrieved May 30, 2018 from <https://www.meteor.com/>
- [5] MongoDB. 2018. MongoDB. Retrieved May 30, 2018 from <https://www.mongodb.com/>
- [6] Task Group on Information Technology Curricula. 2017. *Information Technology Curricula 2017: Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology*. ACM, New York, NY, USA.
- [7] Oracle. 2018. MySQL. Retrieved May 30, 2018 from <https://www.mysql.com/>
- [8] Pluralsight. 2018. CodeSchool. Retrieved May 30, 2018 from <https://www.codeschool.com/> CodeSchool was purchased by Pluralsight. As of June 1, 2018, all content on codeschool.com will be migrated to pluralsight.com.
- [9] Edward Sobiesk, Jean Blair, Gregory Conti, Michael Lanham, and Howard Taylor. 2015. Cyber Education: A Multi-Level, Multi-Discipline Approach. In *Proceedings of the 16th Annual Conference on Information Technology Education (SIGITE '15)*. ACM, New York, NY, USA, 43–47. <https://doi.org/10.1145/2808006.2808038>
- [10] United States Military Academy 2018. *Educating Army Leaders: Developing Intellect and Character to Navigate a Diverse and Dynamic World*. United States Military Academy, West Point, NY, USA. Retrieved May 23, 2018 from https://www.usma.edu/strategic/SiteAssets/SitePages/Home/Educating%20Army%20Leaders_Jan%202018.pdf