INFORMATION SYSTEMS EDUCATION JOURNAL

Volume 19, No. 3
June 2021
ISSN: 1545-679X

In this issue:

4. Development of a Small Cybersecurity Program at a Community College
Patrick Ward, Claremont Graduate University

11. Cognitive Learning Strategies in an Introductory Computer Programming Course
Pruthikrai Mahatanankoon, Illinois State University
James Wolf, Illinois State University

21. An Inventory of Privacy Curricula Offerings in Higher Education Student
Dustin Steinhagen, Dakota State University
Chase Lucas, Dakota State University
Mary Francis, Dakota State University
Mark Lawrence, Dakota State University
Kevin Streff, Dakota State University

31. Online teaching effectiveness: A case study of online 4-week classes in a graduate information systems program
Joni K. Adkins, Northwest Missouri State University
Cindy Tu, Northwest Missouri State University

38. The Importance of Faculty/Staff Support During Times of Crisis
Kiku Jones, Quinnipiac University
Bruce Saulnier, Quinnipiac University
Julia Fullick-Jagiela, Quinnipiac University
Lori N. K. Leonard, The University of Tulsa

47. From Engagement to Empowerment:
Project-Based Learning in Python Coding Courses
Mark Frydenberg, Bentley University
Kevin Mentzer, Bryant University

iscap
Information Systems & Computing Academic Professionals

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published by **ISCAP** (Information Systems and Computing Academic Professionals). Publishing frequency is six times per year. The first year of publication was 2003.

ISEDJ is published online (https://isedj.org). Our sister publication, the Proceedings of EDSIGCON (https://proc.iscap.info) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the EDSIGCON conference. At that point papers are divided into award papers (top 15%), other journal papers (top 25%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 40%.

Information Systems Education Journal is pleased to be listed in the Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org. Special thanks to members of ISCAP/EDSIG who perform the editorial and review processes for ISEDJ.

INFORMATION SYSTEMS EDUCATION JOURNAL

Editor's Note – ISEDJ 2021

Paul Witman
Incoming Senior Editor, ISEDJ
editor@isedj.org

School of Management
California Lutheran University
60 W. Olsen Road
Thousand Oaks, CA 91360 USA

Please join me in a heartfelt Thank You to Dr. Jeffry Babb, outgoing Senior Editor of the Journal. Dr. Babb took on the editorship in January of 2016 and has led the Journal well in his time as Editor, supporting the mission of publishing the best submissions to the EDSIGCON conference on Computing Education each year.

It is my pleasure to take the helm of the Information Systems Education Journal. I have been involved with the Journal in one way or another since 2010, including serving as Cases Co-Chair and Editor for the last few years, with Dr. Anthony Serapiglia and Dr. Ira Goldstein. A wonderful and much-appreciated community of scholars comes together each year to create the ISEDJ content, and I appreciate your good work.

The Journal's pages are filled with the best works from the EDSIGCON conference and its Teaching Cases track. There are papers from each of the various tracks that make up the Conference, so everyone will find something of interest this year. EDSIGCON is a great conference at which to learn and share ideas about teaching the Information Systems discipline. If you have not attended, or not attended recently, I encourage you to consider writing a paper for the Conference, and to get involved in supporting its mission. Like any conference, it can always use more reviewers and other volunteers to produce a great event.

Your editorial board and I look forward to meeting both prior and new authors and reviewers at the 2021 Conference in Washington, D.C. this November. At the same time, you will have the opportunity to attend and present at the Conference on Information Systems Applied Research (CONISAR). Check out the Conferences' web sites at www.edsigcon.org and www.conisar.org.

Many thanks also to the Journal's Associate Editors, Dr. Anthony Serapiglia and Dr. Jason Sharp. Their input into final editing decisions is vital to our editorial process, and I appreciate their contributions and enthusiasm. Finally, thank you to all the volunteer reviewers of the Editorial Board, both for the Cases special issue and for the remainder of this Volume.

We will work together to set goals for the Journal for the years ahead, including gathering input from the ISCAP Board of Directors and editorial board members. I anticipate that among those goals will be to increase the ranking and reputation of the Journal, and to strengthen our editorial processes.

I look forward with you to a great Conference, to collegial networking, and to putting together another high-quality collection of scholarship focused on Information Systems Education. Thank you in advance for your contributions!

Development of a Small Cybersecurity Program at a Community College

Patrick Ward
patrick.ward@cgu.edu

Center for Information Systems and Technology
Claremont Graduate University
Claremont, CA 91711, US

Abstract

This paper introduces the problem of constructing a methodology to develop a cybersecurity program. The goal of the program is to prepare students graduating from an accredited two-year college for success in cybersecurity careers. Several challenges must be addressed such as program accreditation, workforce development, pedagogy, existing curriculum standards, and the process to achieve a Department of Homeland Security/National Security Agency Center of Academic Excellence in Cyber Defense (CAE-CD) designation. All of these serve as inputs in constructing a methodology to develop a program to meet local industry needs for cyber professionals.

Keywords: Cyber Security Education, Curriculum Development, Pedagogy

1. INTRODUCTION

This paper seeks to offer guidelines to faculty and staff in building a cybersecurity curriculum for a two-year community college by reporting how a community college has been developing a small cybersecurity program since the fall 2016 semester and discusses the motivation for various changes made as the program has evolved over the last 3 years. This paper shows how the curriculum was adapted to meet various challenges. This paper also addresses how the college has changed course delivery due to campus shutdown. The original program was an Association to Advance Collegiate Schools of Business (AACSB) accredited business information systems program, and this program needed to be changed to both accommodate local industry's evolving need for cybersecurity professionals and to be accredited with Association of Technology, Management, and Applied Engineering (ATMAE). The program also needs to serve the growing demand for cybersecurity professionals nationwide (Coulson, Mason, & Nestler, 2018) (Burning Glass Technologies, 2019). The new cybersecurity program has been developed from the original because (1) it was cost effective to do so, (2) existing faculty could be used to start the program, and (3) the faculty wished that the program retain its AACSB accreditation. The third goal was later determined to be untenable as the AACSB accreditation requirements changed. These three constraints shaped the curriculum development. This paper discusses the changes made to the original program to support the new cybersecurity program and explains why each change was made.

The Business and Information Technologies (BUS) division at the author's community college (CC) had an information systems technology program since 2009. Information systems is generally considered a business school competence, (Devece Carañana, Peris-Ortiz, & Rueda-Armengot, 2016), and, as information technology has evolved to be more of an engineering disciple, therefore, it was necessary to create a new program and move it to a new division to make the program independent of the BUS division to satisfy the IT needs that a new cybersecurity program needed to fill. Before the fall 2016 semester, the decision was made to create a new Computer Information Technology

(CIT) department housed within the Engineering and Information Technology (E&IT) division.

There are numerous needs for a cybersecurity program, among them were: (1), no existing 2-year cybersecurity program within commuting distance of the local metropolitan area, (2) a need for cybersecurity professionals across many industries, and (3) a local need for cybersecurity professionals as many of the college's graduates are employed within the local area. Faculty and administration considered each of the above needs before deciding to create the new cybersecurity program.

The local public 4-year university offers a concentration in cybersecurity that is oriented towards educating cybersecurity professionals to be employed outside the local area, thus the faculty determined that the CC would offer a program that specifically trained cybersecurity technicians needed locally. However, the CC also recognized the need to keep its graduates employable nationally, so the CIT department sought to align the new cybersecurity program with both Center of Academic Excellence in Cyber Defense (CAE-CD) guidelines (National Security Agency (NSA), 2018) and industry recognized certifications (CompTIA, 2016).

Initially the new cybersecurity program contained business and accounting courses to meet its AACSB accreditation standards. Prior to the fall 2017 semester, the program eliminated those courses and added a natural science course to meet ATMAE accreditation standards. The core CIT department course requirements include networking, systems analysis, database concepts, Linux, and programming to support three concentrations: networking, programming, and cyber defense. New courses were added to support the cyber defense concentration including ethical hacking/penetration testing, firewalls, forensics, network security, and an introduction to information assurance. In all, 7 new courses totaling 21 units were added.

New faculty were hired to teach the additional 7 new courses. The author was among the first new hires to meet this need. The 7 new courses were each chosen to meet needs expressed by local industry. The challenge presented to the faculty was to align the courses both with industry needs and industry-recognized certifications to provide value to the students and to the local and national employers.

The rest of this paper is structured as follows: section 2 is the literature review, section 3 is the discussion, section 4 is the summary, section 5 is the conclusion with the references in section 6.

2. LITERATURE REVIEW

The Association of Computing Machinery (ACM) released the first set of curriculum guidelines for cybersecurity programs for 2-year colleges in 2020 (Tang, Tucker, Servin, Geissler, & Stange, 2020). This provided the long-awaited mapping to the CAE knowledge units (KUs) and a mapping of the competencies into the NICE framework.

What follows is a brief literature review of the various efforts to define a cybersecurity curriculum.

Many of the efforts focus on curriculum design for ABET accreditation for a 4-year degree program: (Mattord & Whitman, 2004); (Smith, Koohang, & Behling, 2010); (Cheung, Cohen, Lo, & Elias, 2011); (Conklin, Cline, & Roosa, Re-engineering Cybersecurity Education in the US: An Analysis of the Critical Factors,, 2014); (Ekstrom, Lunt, Parrish, Raj, & Sobiesk, 2017); (Knapp, Maurer, & Plachkinova, 2017); (Dawson, Wang, & Williams, 2018); (de Leon, Jillepalli, House, Alves-Foss, & Sheldon, 2018); (Raj & Parrish), but few described undergraduate 2-year programs applying for ATMAE accreditation (Doggett, 2015). One early effort by (Bacon & Tikekar, 2003) attempted to create an information assurance curriculum. (McGinnis & Comstock, 2003) attempt to integrate the NICE framework into a curriculum. Another early effort by (Bogolea & Wijekumar, 2004) described an effort to form a security curriculum from various technology courses. (Dennis, El-Gayar, & Streff, 2004) describe an effort to create a curriculum based on NISTISSI-4011 standards. Both (Schweitzer, Humphries, & Baird, 2006) and (Clark & Stoker, 2018) discuss the process of achieving a CAE designation for a curriculum. (Conklin & Bishop, 2018) do a thorough job of comparing the CSEC2017 (ACM, IEEE-CS, AIS SIGSEC, and IFIP WG 11.8, 2017) curriculum standards with the CAE designation requirements.

Recently, (Costigan & Hennessey, 2016) released a generic reference cybersecurity curriculum for NATO. While the curriculum does focus on national security, its risk-orientation is applicable across many industries. The NATO curriculum emphasizes international cybersecurity organizations, policies, and standards, so it is oriented towards the compliance area of cybersecurity. The curriculum addresses risk management applicable to this author's proposed curriculum.

(Conklin, 2018) proposes 3 new core knowledge units (KUs) for a cybersecurity curriculum. His proposal is based on standard accreditations such as ABET and ATMAE, and on specific curriculum guidelines like CS2008 (ACM and IEEE Computer Society, 2008) and CSEC2017, and on specific industry certifications from CompTIA. He does not address specific industry certifications from organizations like $(ISC)^2$ and EC-Council, which are addressed in (Knapp, et. al., 2017). His proposal includes cybersecurity principles, fundamental concepts, and IT Systems components. The IT Systems components address areas tested by attaining industry certifications from CompTIA. The principles and fundamental concepts are addressed by tests from organizations like $(ISC)^2$.

(Furnell, S., Michael, K., Piper, F., Chris, E., Catherine, H., & Ensor, C., 2018) discuss the national cybersecurity program from the UK's National Cyber Security Centre (NCSC). Initially the program was developed from the CS2013 (ACM, 2013) and later validated by the CSEC2017 curriculum guidelines. Furnell chose to use the Institute of Information Security Professionals' (ISSP) Skills Framework (Institute of Information Security Professionals, 2010) as a starting point to develop a curriculum. This presented challenges when attempting to integrate industry input into curriculum design as the ISSP's focus is on security management and not on the technical skills in which employers need to have graduates trained. The paper does acknowledge the CSEC2017 effort, which also shapes this author's proposed curriculum.

(Harris & Patten, 2015) use learning theory from Bloom's (Bloom & Krathwohl, 1956) and Webb's (Webb, 1997) taxonomies and student learning outcomes to add topics and to create new courses in an existing ABET-accredited curriculum. The authors also provide a useful mapping of curriculum topic areas and examples of student work. This was the first paper that mapped both the curriculum topic areas and courses to learning outcomes including examples of work that students did to achieve them. I map these topic areas and examples to existing courses in section 3.

(Kim & Beuran, 2018) propose a model for educational program design methodology that incorporates many of the inputs referenced in the preceding paragraphs. Their model attempts to incorporate all stakeholders, as in the UK and NATO models, including industry, who ultimately employs the graduates of these programs. Their model also incorporates the changing nature of cybersecurity by proposing that new courses and/or existing courses be modified to accommodate emerging technologies. Finally, their model acknowledges that program development starts with a review of existing programs and pedagogical method selection.

In the discussion, this author will apply a modified Kim & Beuran model to develop a proposed cybersecurity curriculum incorporating a few more inputs.

3. DISCUSSION

The effort to change the program and the curriculum is ongoing at a community college since the fall 2016 semester. As the 2016-2017 academic year progressed, AACSB changed its accreditation requirements necessitating that more business courses be added to the new program's curriculum. Since CIT decided to remain within E&IT, the department elected to seek a new program accreditation that was more aligned with the rest of the E&IT division programs. The decision was made to seek ATMAE (ATMAE, 2019) accreditation for the new program because although many of the programs in the E&IT division were ABET-accredited (Accreditation Board for Engineering and Technology, Inc., 2019), there was no ABET accreditation available for a 2-year cybersecurity program at that time.

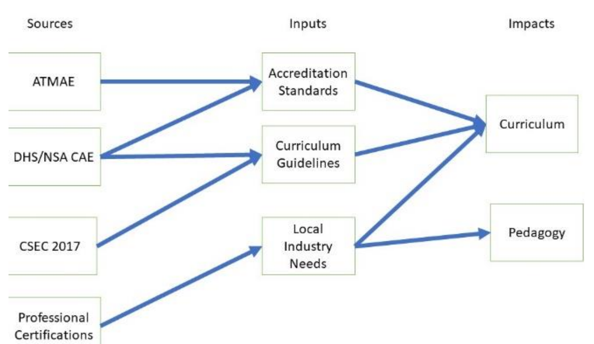

Figure 1. Program Development Inputs

The program has been steadily increasing in enrollment from 20 in the Fall 2016 semester to 40 in the Fall 2019 semester. AACSB standards were replaced with ATMAE standards for accreditation. Local industry is consulted twice yearly for their inputs regarding the program and for suggestions for improvement. Various certification organizations are reviewed for the different certifications offered, their relevance to the program, and local industries' desire for them. The proposed framework with the inputs is specified in Figure 1.

New and Modified Courses
New courses need to be added to the curriculum to accommodate local industry needs and emerging technologies. New courses are offered for two years to assess their effectiveness before they are added to the curriculum. This allows the college to flexibly adapt to local industry needs. Two courses adapted to meet industry needs were (1) digital forensics and (2) penetration testing and network defense. The digital forensics course was adapted to meet local industry needs by providing a more comprehensive foundation for students to be ready to be trained by future employers or to take graduate courses. The penetration testing and network defense course was adapted to cover topics like malware analysis using data analytics and a brief introduction to Python programming. Future courses may include topics like cloud computing, data analytics, and mobile computing. A Special Topics in CIT course was added to accommodate some of the changing trends in the industry, for example, cryptocurrency and Internet of Things (IoT). The examples above illustrate that the ability for curriculum designers to be able to add new courses and modify existing courses is essential to remaining current with industry.

Course Sequencing
Course sequencing is also an issue for several reasons. Notably, the course prerequisites need to be redefined to ensure that students are at least exposed to the concepts in one course prior to applying them in subsequent courses. Another factor that needs to be overcome is the students' reluctance to retain information from one course to apply in another course. Initially, students take courses that depend on Linux knowledge before they take the Introduction to Linux course. The students are also expected to understand basic programming concepts before they take courses involving scripting, a topic covered in the Introduction to Linux course. The students' application of shared concepts is most apparent in the network security course where the students engage in undergraduate research to prepare a paper and a presentation to their peers across the college as part of a student research symposium. The network security course assumes that the students have been exposed to both software and network security issues in Computer Science 1, introduction to Networking, and Principles of Information Assurance. The examples above illustrate how to determine the course prerequisites necessary for the students to begin to master to be able to learn new material.

Course Delivery
Course delivery is also challenging as it requires the campus IT group to set up a firewalled classroom/lab environment in which the students could freely practice the techniques they learned. This setup does not provide a satisfactory solution for students unable to come to the classroom, so a cloud-based solution is under consideration until the spring break when a campus shutdown shifted the faculty's priorities. The college elected to extend the spring break for one week to allow the faculty to investigate alternatives to enable teaching online. The result was a pedagogy consisting of a combination of a flipped classroom and a tutorial-style approach. The idea is to have students come to class with their homework problems, and the instructor would be available to help the students help each other.

This author elected to move two cyber security courses: 1) network security, with 19 students, and 2) penetration testing, with 4 students, completely online. The campus IT group had not setup the firewalled classroom/lab environment, so the program used an online environment from the textbooks' publisher until the end of the spring semester. This author held synchronous video conferenced classroom lab sessions where students could connect via screen sharing to work either singly or in groups with the instructor to work on the assigned laboratory exercises. This provided a rare opportunity for the students to collaborate with the instructor's guidance that was not previously available in the conventional on-ground lecture style. Course delivery depends on subject and with the changing needs of colleges to move more online, these methods will change accordingly. The examples above suggest that helping students work through lab exercises in real-time class sessions may be beneficial.

Industry Standard Alignment
The courses are also aligned with various industry-recognized certifications so that graduating students are able to attain certifications to make them more employable by both the local industry and nationally. Every curriculum developer can benefit from being aware of both local industry needs and industry-recognized certifications when developing or revising a curriculum. Currently, faculty are aligning course material with industry-recognized certifications to make graduates more attractive to employers. Table 1 lists only the computer information technology courses and their

associated industry-recognized certifications in the current program curriculum.

Term/Year	Course	Course Name
Fall/1st	CISP 1010	Computer Science 1
	CITC 1302	Introduction to Networking CompTIA Network+
	CITC 1351	Principles of Information Assurance
Spring/1st	CISP 1020	Computer Science 2
	CITC 1303	Database Concepts
	CITC 1332	UNIX/Linux Operating System CompTIA Linux+
	CITC 2326	Network Security CompTIA Security+
Fall/2nd	CITC 2335	Systems Analysis and Design
	CITC 2352	Digital Forensics
	CITC 2357	IoT Security
	CITC 2363	Internet Intranet Firewalls and eCommerce
Spring/2nd	CITC 2354	Advanced Digital Forensics
	CITC 2356	Penetration Testing and Network Defense CompTIA PenTest+
	CITC 2391	Special Topics in CIT
	CITC 2399	CIT Internship

Table 1. Proposed Program Curriculum

4. SUMMARY

This paper seeks to offer guidelines to faculty and staff in building a cybersecurity curriculum for a two-year community college. Regardless of the institution, the same issues: program accreditation, workforce development, pedagogy, existing curriculum standards and CAE-CD designation need to be addressed. Although the ATMAE program accreditation requirements are not the same as they are for ABET, the same process of applying the standards is used. The contribution here related to the DHS/NSA CAE-CD KUs is also equally applicable to the ABET knowledge, skills, and abilities (KSAs) and to the recently released Cyber2yr2020 guidelines (Tang, et. al., 2020), which are, in turn, mapped to both NICE and CAE recommendations.

This case study was limited to the workforce development needs of the local industry. The local firms range from small businesses to somewhat larger employers in various industries from manufacturing to health insurance. Although there are no immediate federal government contract employers in the area, the curriculum standards used are equally applicable.

The limitations on this case study are that they are specifically relevant to a two-year community college cybersecurity program seeking both a DHS/NSA CAE-CD designation and ATMAE program accreditation. Four-year universities have the option of seeking program accreditation with ABET. The NICE framework serves as a guideline to meet the DHS/NSA CAE-CD requirements for the designation, but a college also needs to have their programs accredited to attract, retain, and place students in industry.

The additional pedagogical challenge of having to convert conventional lab/lecture sessions to a completely online format was also met in the spring semester. The students benefited greatly from the experience per their course exit surveys including some who stated that the online exercises helped them understand the material better. This was additionally validated by their exam scores improving after the switch to the online collaborative environment, and the students' final exam scores being better than that of the previous semester's students. The courses that were modified in the spring semester will be modified again this coming spring to take advantage of the opportunities of teaching online. For the fall, one more course: Internet Intranet Firewalls and eCommerce will be modified to be taught online.

5. CONCLUSION

As ubiquitous connectivity has infiltrated our lives, it is now more important to defend ourselves from the myriad of cyberthreats. We need more and better-educated cybersecurity professionals to defend us. The recent campus shutdown presented a new challenge to instructors attempting to educate students to prepare them to be cybersecurity professionals. This paper is an attempt to provide institutions of higher learning guidance on developing accredited cyber security programs and give an example of how one two-year institution is developing their program. The lessons learned here are applicable to other two-year programs and to four-year programs looking to either start or revise existing programs.

6. REFERENCES

Accreditation Board for Engineering and Technology, Inc. (2019, May 1). ABET Accredited Program Search. Retrieved from

ABET: http://main.abet.org/aps/Accreditedprogramsearch.aspx

Association for Computing Machinery, IEEE Computer Society (IEEE-CS), Association for Information Systems Special Interest Group on Security (AIS SIGSEC), and International Federation for Information Processing Technical Committee on Information Security Education (IFIP WG 11.8), "Cybersecurity Curricula 2017: Curriculum Guidelines for Undergraduate Degree Programs in Cybersecurity," ed, 2017

ACM, "Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science," ed: Association for Computing Machinery (ACM), 2013, p. 518.

ACM and IEEE Computer Society, "CS 2008: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science," ed: IEEE/ACM Joint Task Force on Computing Curricula, 2008.

Association of Technology, Management, and Applied Engineering (ATMAE). (2019). 2019 Accreditation Handbook. Association of Technology, Management, and Applied Engineering (ATMAE), Retrieved from https://www.atmae.org/resource/resmgr/accred_2018/2019_Accreditation_Handbook.pdf

Bacon, T., & Tikekar, R. (2003). Experiences with developing a computer security information assurance curriculum. Journal of Computing Sciences in Colleges, 18(4), 254-267.

Bloom, B., & Krathwohl, D. (1956). Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain. New York, NY, US: Longmans, Green.

Bogolea, B., & Wijekumar, K. (2004). Information Security Curriculum Creation: A Case Study. 1st Annual Conference on Information Security Curriculum Development (pp. 59-65). Kennesaw, GA, USA: ACM. doi:10.1145/1059524.1059537

Burning Glass Technologies. (2019). What’s Trending in Jobs and Skills. Retrieved from Burning Glass Technologies: http://www.burning-glass.com/

Cheung, R. S., Cohen, J. P., Lo, H. Z., & Elias, F. (2011). Challenge based learning in cybersecurity education. Proceedings of the 2011 International Conference on Security & Management, 1.

Clark, U., & Stoker, G. (2018). Reflections on Applying for CAE-CDE Designation. Proceedings of the EDSIG Conference. 2473, p. 3857. Norfolk, VA, USA: Information Systems & Computing Academic Professionals (ISCAP).

CompTIA, "CompTIA Security+ Certification Exam Objectives," p. 28, 2016

Conklin, W. A. (2018, March). What Constitutes Core in a Cyber Security Curriculum? Journal of The Colloquium for Information System Security Education, 5(2), 14-14.

Conklin, W. A., Cline, R. E., & Roosa, T. (2014). Re-engineering Cybersecurity Education in the US: An Analysis of the Critical Factors, 47th Hawaii International Conference on System Sciences (pp. 2006-2014). Waikoloa, HI, USA: IEEE. doi:10.1109/HICSS.2014.254

Conklin, W., & Bishop, M. (2018). Contrasting the CSEC 2017 and the CAE Designation Requirements. 2018 51st Hawaii International Conference on System Sciences (pp. 2435-2441). Waikoloa, HI, USA: IEEE.

Costigan, S., & Hennessey, M. (2016). Cybersecurity: A Generic Reference Curriculum. NATO. Retrieved from https://www.nato.int/nato_static_fl2014/assets/pdf/pdf_2016_10/20161025_1610-cybersecurity-curriculum.pdf

Coulson, T., Mason, M., & Nestler, V. (2018). Cyber Capability Planning and the Need for an Expanded Cybersecurity Workforce, Communications of the IIMA, 16(2). Retrieved from https://scholarworks.lib.csusb.edu/ciima/vol16/iss2/2/

Dawson, M., Wang, P., & Williams, K. (2018). The role of CAE CDE in cybersecurity education for workforce development. Information Technology-New Generations, 127-132.

de Leon, D. C., Jillepalli, A. A., House, V. J., Alves-Foss, J., & Sheldon, F. T. (2018). Tutorials and laboratory for hands-on OS cybersecurity instruction. Journal of Computing Sciences in Colleges, 34(1), 242-254.

Dennis, T., El-Gayar, O. F., & Streff, K. (2004). model program in information assurance and computer security. IACIS International Association for Computer Information Systems, 4(2), 97-102.

Devece Carañana, C., Peris-Ortiz, M., & Rueda-Armengot, C. (2016). What are the competencies in information systems required by managers? Curriculum development for management and public administration degrees. Technology Innovations and Education (2), 10. doi:10.1186/s40660-016-0016-2

Doggett, M. (2015). Defining the technology management body of knowledge for ATMAE-accredited programs. Technology Interface International Journal, 16(1), 87-99.

Ekstrom, J. J., Lunt, B. M., Parrish, A., Raj, R. K., & Sobiesk, E. (2017). Information technology as a cyber science. Proceedings of the 18th Annual Conference on Information Technology Education (pp. 33-37). ACM. doi:10.1145/3125659.3125697

Furnell, S., Michael, K., Piper, F., Chris, E., Catherine, H., & Ensor, C. (2018). A national certification programme for academic degrees in cyber security. IFIP World Conference on Information Security Education (pp. 133-145). Springer Cham. doi:10.1007%2F978-3-319-99734-6_11

Harris, M. A., & Patten, K. P. (2015). Using Bloom's and Webb's Taxonomies to Integrate Emerging Cybersecurity Topics into a Computing Curriculum. Journal of Information Systems Education, 26(3), 219-235.

Institute of Information Security Professionals: IISP information security skills framework, V6.3, July 2010. Institute of Information Security Professionals (2010)

Kim, E., & Beuran, R. (2018). On designing a cybersecurity educational program for higher education. Proceedings of the 10th International Conference on Education Technology and Computers (pp. 195-200). ACM. doi:10.1145/3290511.3290524

Knapp, K., Maurer, C., & Plachkinova, M. (2017). Maintaining a cybersecurity curriculum: Professional certifications as valuable guidance. Journal of Information Systems Education, 28(2), 101-113.

Mattord, H. J., & Whitman, M. E. (2004). Planning, building, and operating the information security and assurance laboratory. Proceedings of the 1st annual conference on Information security curriculum development (pp. 8-14). Kennesaw: Kennesaw State University.

McGinnis, D. R., & Comstock, K. (2003). The implications of information assurance and security crisis on computing model curricula. Information Systems Education Journal, 1(9), 1-12.

National Security Agency (NSA). (2018). National Centers of Academic Excellence (CAE) Resource Guide. National Security Agency (NSA). Retrieved from https://niccs.us-cert.gov/sites/default/files/documents/pdf/cae_program_guidance.pdf?trackDocs=cae_program_guidance.pdf

Raj, R. K., & Parrish, A. (n.d.). Toward standards in undergraduate cybersecurity education in 2018. Computer, 51(2), pp. 72-75.

Schweitzer, D., Humphries, J., & Baird, I. (2006). Meeting the criteria for a Center of Academic Excellence (CAE) in information assurance education. Journal of Computing Sciences in Colleges, 22(1), 151-160.

Smith, T., Koohang, A., & Behling, R. (2010). Formulating an effective cybersecurity curriculum. Issues in Information Systems, 11(1), 410-416.

Tang, C., Tucker, C., Servin, C., Geissler, M., & Stange, M. (2020). Curricular Guidance for Associate-Degree Cybersecurity Programs. Proceedings of the 51st ACM Technical Symposium on Computer Science Education (pp. 1285-1285). ACM. Retrieved from http://ccecc.acm.org/files/publications/Cyber2yr2020.pdf

Webb, N. (1997). Research Monograph no. 6: Criteria for Alignment of Expectations and Assessments in Mathematics and Science Education. Washington, DC, US: Council of Chief State School Officers.

Cognitive Learning Strategies in an Introductory Computer Programming Course

Pruthikrai Mahatanankoon
pmahata@ilstu.edu

James Wolf
jrwolf@ilstu.edu

School of Information Technology
Illinois State University
Normal, IL 61761, USA

Abstract

Learning a computer programming language is typically one of the basic requirements of being an information technology (IT) major. While other studies previously investigate computer programming self-efficacy and grit, their relationships between "shallow" and "deep" learning (Miller et al., 1996) have not been thoroughly examined in the context of computer programming. Exploratory factor analyses using data collected from undergraduate information technology students, who just completed their first programming class shows distinct shallow and deep learning in computer programming. While shallow learning supports previous research, deep learning has three sub-scale activities: practice by examples, analytical thinking, and diagramming. The results also reveal that computer programming grit and self-efficacy have low to moderate correlations with shallow and deep learning, requiring further examination. Preliminary regression analyses also find that shallow learning positively influences computer programing grit and self-efficacy. Shallow learning strategies may be more widely employed during the initial stages of computer programming, while deep learning strategies may be more prevalent in higher-level computer programming courses. IT educators can examine this shift in strategies by observing students as they progress from introductory to advanced computer programming courses.

Keywords: Cognitive Learning, Programming, Deep Learning, Grit, Self-efficacy, Shallow Learning

1. INTRODUCTION

After over 50 years of study, the low success rates in introductory programming courses remain among the most intractable problems in computing education. The problem has been widely studied, but solutions have proven elusive. The lack of success in introductory programming courses and aversion to computer programing are often mentioned as significant factors in low retention numbers (McGettrick et al., 2005).

In addition to the large number of students with low performance, instructors often report a nearly equal number of students with high performance (Robins, 2010). As a result, instructors in introductory programming courses often report a binomial or two-humped distribution, with students grouped into the left and right tails, and few in the middle. This has led to a great deal of work that aims to understand the difference between these two student groups.

This reported binomial distribution has also led to a belief that programming is more innate than other academic subjects. That is that some people were born to be programmers while others are not. This belief is best captured in what Lister (2010) calls the idea of the "geek gene." According to this theory, those students born with the "geek gene" have the innate ability to program and the attitude necessary to succeed in programming courses. Other students, presumably those born without the "geek gene", are genetically predestined to fail.

However, there is little empirical evidence to support this hypothesis. While most research finds that there is a correlation between mathematical aptitude and success in introductory programming courses, strong math skills are correlated with overall collegiate success (Pea and Kurland, 1984). Further programmer aptitude tests, like the PAT (Programmer Aptitude Test) administered by IBM and others, have shown little association with career programming success (Pea and Kurland, 1984). As a testament to its lack of predictive power, IBM no longer administers the PAT to prospective programmers (Lorenzen and Chang, 2006).

If innate skills and aptitude tests have little empirical evidence, then what could be the determinants of success in an introductory computer programming course? Recently researchers have begun examining cognitive factors and traits associated with student success in programming courses that are malleable and can be taught. For example, both grit and computer self-efficacy are associated with student success in programming courses (Wolf and Jia, 2015; Kanaparan, Cullen, & Mason, 2013; and Rex & Roth, 1998) are malleable (McClendon et al., 2017; Bandura, 1997). Similarly, student choice of cognitive learning strategies (deep or shallow) has been found to affect academic success and is changeable (Marton & Säljö, 1976b).

There has been limited research on the impact of grit, self-efficacy, and cognitive learning strategies on student success in introductory programming courses. This work will examine the relationship between these three constructs using scales developed explicitly for the task of computer programming. The study's purposes are 1) to test the measurement of cognitive learning strategies, i.e., shallow and deep learning, in the computer programming settings, and 2) to understand their relationships with grit and self-efficacy. Examining these relationships will help us extend the roles of grit and self-efficacy to students' cognitive learning strategies.

2. RELATED LITERATURE

Coding Grit

Grit is the trait-level perseverance and passion needed to obtain long-term goals (Duckworth et al., 2007). Grit is associated with academic success in a variety of academic settings (e.g., Duckworth and Quinn, 2009; Duckworth et al., 2007; & Strayhorn, 2013). Grit changes over time. People become gritter as they age, and their grit can be strengthened with deliberate practice (McClendon et al., 2017).

Grit has been widely studied in academic settings. For example, Duckworth et al. (2007) found that "grittier" students were more likely to succeed in both an Ivy League university and the United States Military Academy. Strayhorn (2013) found that African American males with higher grit earned higher grades in college than same-race male peers with lower grit. Similarly, Wolf and Jia (2015) found that grittier students earned higher grades in introductory programming courses than their less gritty peers.

While an abundance of studies demonstrates the positive link between intelligence and academic achievement (e.g., Laven, 1965), Duckworth et al. (2007) suggest that grit may be a better predictor of student success than talent or intelligence. Similarly, Wolf and Jia (2015) found that grit was a more powerful predictor of success in programming courses than college entrance exam scores.

While Wolf and Jia (2015) investigated the relationship between "generic grit" and student programming success, Mahatanankoon & Sikolia (2017) and Mahatanankoon (2018) altered the 12-point grit scale (Duckworth et al., 2007) to capture computer programming specific grit or coding grit. Mahatanankoon (2018) defined coding grit as the ability to "persevere and focus through computer programming activities." Mahatanankoon (2018) found that female computer science/information students were grittier. That is, they had higher levels of perseverance and long-term interest in computer programming than their male counterparts. In related work, Mahatanankoon & Sikolia (2017) found that passion and grit were positively correlated with computer programming attitude and retention in computer majors.

Computer Programming Self-efficacy

The widely studied information systems construct, computer self-efficacy, is an adaption of the more general self-efficacy (Compeau et al., 2006). Self-efficacy is four sources and reflects a future-oriented belief about one's ability to execute a specific task in a given context (Bandura, 1997). For example, computer self-efficacy is one's belief about their ability to use a computer (Compeau & Higgins, 1995). The four sources of self-efficacy beliefs are performance accomplishment, vicarious experience, verbal persuasion, and physiological states (Bandura, 1977). As a result, self-efficacy is malleable.

Computer self-efficacy (CSE) is positively correlated with academic success in computer programming courses (e.g., Kanaparan, Cullen, & Mason, 2013 and Rex & Roth, 1998). Students with higher computer self-efficacy are more comfortable using computers and more confident in their computer-related skills. It is not surprising that this comfort and confidence leads to higher grades in computer programming courses.

In the seminal work in this area, Compeau & Higgins, 1995 defined computer self-efficacy (CSE) as the judgment of one's capabilities to use a computer in diverse situations. Mahatanankoon (2018) adapted the computer self-efficacy (CSE) (Compeau & Higgins, 1995) scale to capture self-efficacy for the specific task of computer programming. Computer programming self-efficacy or coding self-efficacy is one's belief about his/her computer programming ability. Mahatanankoon (2018) found that computer programing grit was a significant predictor of programming self-efficacy.

Deep and Shallow Learning Approaches
Marton, F., & Säljö, R. (1976a) identified two cognitive learning strategies: deep and surface. Within this framework, students adopt deep learning strategies when they intend to fully understand the subject matter and link it to their prior knowledge and personal experiences. In contrast, students adopt shallow learning strategies when their intention is merely to reproduce information without any further analysis (Murphy & Tyler, 2005).

Students using surface cognitive strategies are primarily concerned with storing the information into short-term memory, they focus on memory strategies (i.e., rote processing, repetition, reciting, and highlighting) (Boyle, Duffy, & Dunleavy, 2003). Students using surface cognitive strategies often rush through assignments and write down the first answer that comes to mind (Anderman, 1992).

Students using deep cognitive processing strategies try to understand new concepts by connecting new material with previously learned material, adopting a critical attitude towards information, and stopping to think about their work (Murphy & Tyler, 2005; Weinstein & Mayer, 1986). Students using deep cognitive processing strategies often monitor comprehension through self-quizzing, and engage in paraphrasing or summarizing (Anderman, 1992). In studies examining student achievement, several have found that academic performance was influenced positively by deep processing (e.g., Fenollar et al., 2007; Cano, 2005; Elliot et al., 1999; Miller et al., 1996).

As with earlier studies, we believe that the constructs under investigation, i.e., coding grit, coding self-efficacy, and cognitive processing, are related. The next section describes our methods.

3. METHODS

We conducted a field study to examine the relationships between coding grit, coding self-efficacy, and student cognitive learning strategies in introductory programming courses. The data were collected in fall 2019 (Sample 1) and spring 2020 (Sample 2). We used Sample 2 to verify Sample 1's results.

Sample 1
In fall 2019, we collected data from information technology students enrolled in systems analysis and design class, which had introductory Java programming as its prerequisite. Introductory Java programming is required for all IT majors in our department. We had 85 initial responses. After eliminating non-IT majors (n=2), telecommunication management (n=4), graduate MSIS students (n=13), duplications (n=6), and incomplete responses (n=7), we had a final total of 53 respondents in the study with Computer Science (32%), Information Systems (42%), and Cyber Security (26%). ANOVA also showed no significant mean differences in the research variables among the IT majors, i.e., Computer Science (n=17), Information Systems (n=22), and Cyber Security (n=14). Male students made up the majority of respondents (74%) in this sample. However, independent t-tests showed no significant mean difference in the research variables between male and female students.

Measures
For computer programming grit and computer programming self-efficacy, we will use previously developed scales (Mahatanankoon & Sikolia, 2017; Mahatanankoon, 2018). To examine shallow and deep learning, we modified the items developed by Greene and Miller (1993) to fit the context of computer programming. Appendix A shows the list of our modified questionnaire.

Analyses and Results of Sample 1
Exploratory factor analysis (EFA) was used to examine the dimensions of shallow and deep

learning in computer programming. Our initial factor analysis indicated five factors with eigenvalues higher than or equal to 1.0. Any factor with cross-loadings was eliminated. Through several iterations of EFA, Table 1 shows a three-factor solution with at least .50 loading value. The solution accounts for 62.7 percent of the total variance.

We see that shallow learning loaded into a single factor (Factor 1, SL1-SL4, DL11). We define *Factor 1* as the set of "fundamental" skills in computer programming, plus DL11 added to the factor. Deep learning skills loaded into two factors. *Factor 2* included DL3 and DL4 as the predominant items. DL3 involves working on several programming examples by repeating the same type of problems. Similarly, DL4 also relies on practicing and checking one's understanding of "new" concepts and rules. *Factor 3* entails analytical thinking of programming, which demands the ability to classify (DL6), analyze (DL7) coding problems, and eventually leading to an optimized solution.

Items	Factor 1	Factor 2	Factor 3
SL1	.636		
SL2	.971		
SL3	.699		
SL4	.540		
DL3		.618	
DL4		.988	
DL6			.917
DL7			.567
DL11	.590		

Table 1: Sample 1 Three-factor Solution

Sample 2

To verify our previous results, we collected another set of data using the students taking systems analysis and design, introduction to software and hardware concepts, and database systems in spring 2020. These courses have a Java programming class as a prerequisite. From our initial 62 responses, we eliminated three incomplete and four duplicated responses (n=7). Three graduate students, one telecommunication management, and three non-IT majors were dropped (n=7), giving us a total of 48 responses: Computer Science (n=11), Information Systems (n=17), and Cyber Security (n=20). This sample also has male students as the majority (87.5 percent). ANOVA found no significant mean differences in the research variables among the IT majors.

Analyses and Results of Sample 2

Our initial factor analysis resulted in four factors with eigenvalues higher than or equal to 1.0. Any factor with cross-loadings or with a loading value below .50 was eliminated. Through several iterations of EFA, Table 2 shows a three-factor solution with at least .50 loading value. The solution accounts for 60.6 percent of the total variance.

Our second sample had the same set of "fundamental" skills as our first sample. The results differed from Pilot Sample 1 in two aspects: 1) DL6 loaded onto Factor 2 as its third item, and 2) DL8 loaded with DL7 as Factor 3. These differences seem plausible, suggesting that the classification of programming problems (DL6) coincide with "practicing by examples" (Factor 2), and that finding a practical application could enhance "analytical thinking" (Factor 3).

Items	Factor 1	Factor 2	Factor 3
SL1	.622		
SL2	.870		
SL3	.655		
SL4	.544		
DL3		.867	
DL4		.799	
DL6		.652	
DL7			.665
DL8			.748

Table 2: Sample 2 Three-factor Solution

Earlier, we questioned the learning role of diagramming activities and included D1-4 during our data collection of Sample 2 (see Appendix A). Using the same EFA process with all previous items plus the new diagramming items (i.e., SL1-5, DL1-11, plus D1-4), the initial scree plot and eigenvalues revealed a five-factor solution. After several items were dropped due to low factor loadings (<.50) or cross-factor loadings, the final EFA iteration had a three-factor solution capturing 65.6 percent of the variability in learning. Table 3 shows the results of the factor loadings higher than .50.

Adding the diagramming items forced other deep learning strategies to load onto the same factor. However, without the diagramming items—dropping D3 and D4—the variability decreased to 58.9 percent (a 6.7 percent reduction) with a two-factor solution, as shown in Table 4.

The two-factor solution also persisted when we eliminated the diagram item (D1-D4) from the model (only Sample 2). To verify this result, we force-loaded the same set of items using Sample 1, the loadings were similar to those of Sample 2, except for low factor loadings of DL2, DL6, and DL9.

Items	Factor 1	Factor 2	Factor 3
SL1	.634		
SL2	.903		
SL3	.792		
SL4	.624		
DL2		.566	
DL3		.766	
DL4		.718	
DL6		.757	
DL9		.750	
DL10		.732	
D1		.590	
D3			.936
D4			.785

Table 3: Sample 2 Three-factors Solution with Diagramming Items

Items	Sample 2 (n=48)		Sample 1 (n=53)	
	SL α=.826	**DL** α=.877	**SL** α=.832	**DL** α=.782
SL1	.641		.696	
SL2	.841		.903	
SL3	.880		.766	
SL4	.557		.548	
DL2^		.624^		--
DL3		.706		.639
DL4		.717		.988
DL6^		.801^		--
DL9^		.779^		--
DL10		.757		.523
D1^		.680^		NA
^ = excluded from composite reliability (α) calculation				

Table 4: Comparing Two-factors Solution of Both Samples

From our EFAs and results, shallow learning in computer programming meant that students focus on memorizing the solution or syntax without the tacit understanding of the logical sequences, concepts, and ideas behind coding. We can provide a list of activities constituting of what we called the "Shallow Learning" (SL-CP) in Computer Programming ($\alpha_{sample1}$=.832, $\alpha_{sample2}$=.826) as:

- I try to memorize the steps for solving programming problems presented in the text or in the lecture (SL1).
- When I study for the tests, I review my class notes and look at solved programming problems (SL2).
- When I study for tests, I used solved programming problems in my notes or in the book to help me memorize the 'programming' steps involved (SL3).
- I find reviewing previously solved programming problems to be a good way to study for a test (SL4).

On the contrary, "Deep Learning" in Computer Programming (DL-CP), from our factor analyses, constitutes *practice by examples*, although the learning activities varied between the two samples. Nevertheless, these recurring activities persisted among our respondents ($\alpha_{sample1}$=.782, $\alpha_{sample2}$=.877):

- I work on several programming examples of the same type of problems when studying this class so I can understand the problems better (DL3).
- I work practice programming problems to check my understanding of new concepts or rules (DL4).
- I work on practice programming questions/problems to check my understanding of new concepts or rules (DL10).

Besides, there could be *other* complementing activities for DL-CP. Based on the data, the sub-activities might include

a) DL-A Analytical Thinking ($\alpha_{sample1}$=.601, $\alpha_{sample2}$=.610)

- I classify programming problems into categories before I begin to work them (DL6).
- When I work a programming problem, I analyze it to see if there is more than one way to get the right solution (DL7).
- While learning new programming concepts, I try to think of practical applications (DL8).

b) DL-D Diagramming ($\alpha_{sample2}$=.816)

- I model different program modules or functions using some diagramming techniques (D3).
- I use some diagramming techniques to understand how programming work (D4).
- ▪ Some programming problems can be visualized using diagrams and models (D1).
- ▪ I draw pictures or diagrams to help me solve some programming problems (DL2).

Relationships with Coding Grit and Self-efficacy

On the one hand, self-efficacy is one's ability to control the outcome of a task-specific belief (Bandura, 1977). *Coding self-efficacy* (C-SE) is defined as "*one's belief about his/her computer programming ability" (Mahatanankoon, 2018, p. 2)*. It should enhance one's belief in the success of learning how to write computer programs. The higher the level of one's belief is, the more likely it is for the person to engage in computer programming. On the other hand, grit—a trait related to perseverance, passion, long-term commitment, and interest (Duckworth and Quinn, 2009)—may also be another internal factor driven by one's intention to enhance their knowledge and skills. Coding Grit (C-G) is defined as "one's ability to persevere and focus through computer programming activities" (Mahatanankoon, 2018, p. 2). Coding grit may encourage long-term learning interests in programming leading to both shallow and deep learning strategies. Therefore, to demonstrate nomological validity, we propose that coding self-efficacy (C-SE) and coding grit (C-G) that can be predicted by SL-CP, DL-CP, DL-Analytical Thinking (DL-A), and DL-Diagramming (DL-D, Sample 2 only).

Tables 5 and 6 reveal low correlations among our exploratory factors and the established measures. We see that coding grit and coding self-efficacy are moderately correlated. The cognitive learning strategies (i.e., SL-CP, DL-CP, DL-A, DL-D) are also moderately correlated, supporting the construct validity and suggesting that deep learning in computer programming are multidimensional (also see Tables 1-3).

	SL-CP	DL-CP	DL-A	C-G	C-SE
SL-CP	1				
DL-CP	.35	1			
DL-A	.28	.47	1		
C-G	-.13	-.18	-.12	1	
C-SE	.07	-.16	-.02	.59	1

Table 5: Sample 1 Correlation Matrix

	SL-CP	DL-CP	DL-A	DL-D	C-G	C-SE
SL-CP	1					
DL-CP	.44	1				
DL-A	.15	.48	1			
DL-D	.33	.59	.62	1		
C-G	.22	-.17	.01	.03	1	
C-SE	.18	-.30	-.11	.02	.50	1

Table 6: Sample 2 Correlation Matrix

From Table 7, we also explored the predictive validity (not hypothesized). The regression showed that the SL-CP positively predicted coding grit and coding self-efficacy. DL-CP, on the other hand, negatively predicted coding grit and coding self-efficacy. These significant findings were found only in Sample 2. In both samples, DL-A and DL-D did not influence the dependent variables.

4. DISCUSSION

Our study examined the measurement of shallow and deep learning in computer programming and tested the variable's relationships to coding grit and coding self-efficacy. EFA reveals the similarities of shallow learning in previous studies: route learning emphasized by memorizing and replicating the steps used to solve programming problems.

However, deep learning constitutes a multi-faceted construct. EFA solutions suggest at least three different activities: practicing, analyzing, and diagramming. Solving advanced programming problems calls for various viewpoints, which may be built on both shallow learning and higher cognitive strategies. Both samples yield inconsistent loadings. Future research warrants a larger sample size.

Our data leads us to question the importance of diagrams and models leading to programming solutions. From our factor analyses, the diagramming items (Sample 2) are not significantly loaded, although DL2 ("I draw pictures or diagrams to help me solve some programming problems') and D1 ("Some programming problems can be visualized using diagrams or models") correlated with a deeper level of learning (see Table 4). There are several plausible explanations:

1) The introductory programming class is the prerequisite of systems analysis and design, in which diagramming techniques are introduced. Therefore, diagramming is less valued by students taking programming for the first time.

2) Instructors have not emphasized a clear connection between the phases of analysis and design to the implementation (coding) activities.

3) Diagraming such as a flow chart or decision tree is used to conceptualize the program control statements, which is a *precursor* to introducing program syntaxes themselves. For example, the domain model class diagram assists the development of class definition (coding).

4) Our diagramming items (D1-4) are oversimplified and do not capture a wide variety

of diagramming activities. UML has a different set of modeling techniques that coincide with the different phases of the systems development life cycle.

The positive correlation between coding grit and coding self-efficacy is consistent with previous work. Similarly, shallow and deep learning strategies also have positive correlations among their designated items, but they are quite distinct from coding grit and coding self-efficacy, as our data have shown. Interestingly, deep learning in computer programming has an inverse relationship with coding grit and coding self-efficacy. We offer several explanations:

Firstly, shallow learning is an essential learning strategy to complete the class. Many students who completed an introductory programming course may focus on shallow learning to get a passing grade. Lizzio et al. (2002) find a positive link between a surface approach ("reproducing" approach with less knowledge integration) and a higher GPA among commerce students. Lizzio et al. (2002) posited that the given the narrower vocational focus of commerce courses and the typically employed assessment methods, surface methods, like shallow learning, may be a logical and strategic choice for students to pursue.

Secondly, deep learning strategies may occur in other advanced programming classes. Our samples are students who have just completed their first introductory programming class. Therefore, it is likely that deep learning has not been incorporated into their learning strategy. Computer programming skills may progress through different learning stages: Students use shallow learning to memorize language syntax, flow controls, and compilation steps. As they progress towards more advanced programming classes, evidence of deep learning strategies could be seen, including more substantial evidence of coding grit and coding self-efficacy. The research finding is mixed regarding the relationship between surface processing and academic performance, with most studies finding the relationship as either not statistically significant or negative (Watkins, 2001).

Thirdly, deep learning strategies may differ from one IT major to another, which affects the level of coding grit and coding self-efficacy required. Students usually begin with similar coding skills in the introductory programming course. As they progress to their intended information technology majors (e.g., computer science, information systems, cybersecurity, telecommunication, and others.), their programming needs and skills will adapt to their changing educational focus. Therefore, we may observe different types of deep learning, e.g., analytical thinking and diagramming, that differ across different IT majors. Echo this sentiment, Beattie et al. (1997) suggest that in certain academic situations adopting a surface approach may be advantageous. Fenollar et al. (2007) suggest that that memorization and rote rehearsal might be appropriate for some types of material and exam formats.

Lastly, we collected our samples from various classes and instructors. Student perceptions of the course workload, teaching quality, and fairness of assessment influenced student choices of learning strategies (Lizzio et al., 2002). It is, therefore, possible that other external factors could influence computer programming learning strategies. All in all, we plan to further investigate this phenomenon using data obtained from junior/senior-level undergraduate students.

5. CONCLUSIONS

The most significant contribution of this work is the development, testing and validation of the Deep Learning in Computer Programming (DL-CP) and Shallow Learning in Computer Programming (SL-CP) scales. This work lays the groundwork for further research into the intersection of coding grit, coding self-efficacy and student learning strategy selection in programming courses. The goal of this work is to better understand why some students struggle in programming courses and to equip instructors with the knowledge needed to help these students succeed.

Despite IT researchers' long tradition of modifying scales to fit specific computer-related tasks, previous work in this area has often utilized generic scales, which may fail to capture the important differences between computer programming courses and other IT or general education courses. By creating coding specific scales for deep and shallow learning strategies, this work also provides tools that others investigating the student achievement in computer programming courses may use to better understand the antecedents of student success or failure.

Future work should examine the relationships between coding grit, coding self-efficacy, shallow and deep learning strategies, and student outcomes in both introductory and advanced programming courses. A longitudinal study of how the learning strategies change with increased programming skills may provide pedagogical insights to instructional scaffolding. It may also be fruitful to explore whether student cognitive learning strategies are associated with learning goals and persistence in computer-related majors.

6. REFERENCES

Anderman, E.M. (1992). Motivation and Cognitive Strategy Use in Reading and Writing. In the annual meeting of the National Reading Conference, San Antonio, Texas.

Beattie IV, V., Collins, B., & McInnes, B. (1997). Deep and surface learning: a simple or simplistic dichotomy?. *Accounting Education*, *6*(1), 1-12.

Bandura, A. (1977). Self-efficacy: toward a unifying theory of behavioral change. *Psychological Review*, *84*(2), 191.

Bandura, A. (1997). Efficacy: The Exercise of Control. *Freeman, New York.*

Barrington, K., & Johnson, D. (2005). The relationship between lab attendance and academic performance in a computer information systems course. In Proceedings of ISECON 2005, Vol. 22 (Columbus OH): 3573

Boyle, E. A., Duffy, T., & Dunleavy, K. (2003). Learning styles and academic outcome: The validity and utility of Vermunt's Inventory of Learning Styles in a British higher education setting. *British Journal of Educational Psychology*, *73*(2), 267-290.

Cano, F. (2005). Epistemological beliefs and approaches to learning: Their change through secondary school and their influence on academic performance. *British Journal of Educational Psychology*, 75(2), 203–221.

Compeau, D. R., & Higgins, C. A. (1995). Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly*, 19(2), 189-211.

Compeau, D., Gravill, J., Haggerty, N., & Kelley, H. (2006). Computer self-efficacy. In P. Zhang and D.F. Galletta (Eds.), *Human-computer Interaction and Management Information Systems: Foundations, Advances in Management Information Systems, 225-261.*

Duckworth, A.L., & Quinn, P. D. (2009). Development and Validation of Short Grit Scale (Grit-S). *Journal of Personality Assessment,* 91(2), 166-174.

Elliot, A.J., McGregor, H.A., & Gable, S. (1999). Achievement goals, study strategies, and exam performances: A mediational analysis. *Journal of Educational Psychology*, 91(3), 549–563.

Fenollar, P., Roman, S., & Cuestas, P.J. (2007). University students' academic performance: An integrative conceptual framework and empirical analysis. *British Journal of Educational Psychology*, 77(4), 873–891.

Jones, C. H. (1984). Interaction of absences and grades in a college course. *The Journal of Psychology*, 116(1), 133-136.

Lavin, D.E. (1956). The Prediction of Academic Performance, Russell Sage Foundation, New York.

Lin, G-Y. (2006). Self-efficacy beliefs and their sources in undergraduate computing disciplines: an examination of gender and persistence. *Journal of Educational Computing Research*, 53(4), 540-561.

Lister, R. (2011). Computing education research geek genes and bimodal grades. *ACM Inroads*, *1*(3), 16-17.

Lizzio, A., Wilson, K., & Simons, R. (2002). University students' perceptions of the learning environment and academic outcomes: implications for theory and practice. *Studies in Higher education*, *27*(1), 27-52.

Lorenzen, T., & Chang, H. L. (2006). MasterMind©: a predictor of computer programming aptitude. *ACM SIGCSE Bulletin*, *38*(2), 69-71.

Mahatanankoon, P. (2018). Exploring the Antecedents to Computer Programming Self-Efficacy. In *Proceedings of the 10th International Conference on Advances in Information Technology* (pp. 1-6).

Mahatanankoon, P., & Sikolia, D. W. (2017). Intention to Remain in a Computing Program: Exploring the Role of Passion and Grit. In the 23rd Americans Conference on Information Systems (AMCIS), Boston (pp. 1-10).

Miller, R., Greene, B., Montalvo, G., Ravindran, B., & Nicholls, J. (1996). Engagement in academic work: The role of learning goals, future consequences, pleasing others, and perceived ability. *Contemporary Educational Psychology*, 21(4), 388–422.

Marton, F., & Säljö, R. (1976a). On qualitative differences in learning: I—Outcome and process. *British Journal of Educational Psychology*, *46*(1), 4-11.

Marton, F., & Säljö, R. (1976b). On qualitative differences in learning—ii Outcome as a function of the learner's conception of the task. *British Journal of Educational Psychology*, 46(2), 115-127.

McClendon, C., Neugebauer, R. M., & King, A. (2017). Grit, Growth Mindset, and Deliberate Practice in Online Learning. *Journal of Instructional Research*, *8*, 8-17.

McGettrick, A., Boyle, R., Ibbett, R., Lloyd, J., Lovegrove, G., & Mander, K. (2005). Grand challenges in computing: Education—a summary. *The Computer Journal*, *48*(1), 42-48.

Murphy, S. M., & Tyler, S. (2005). The relationship between learning approaches to part-time study of management courses and transfer of learning to the workplace. *Educational Psychology*, *25*(5), 455-469.

Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New ideas in psychology*, *2*(2), 137-168.

Rex, K., & Roth, R. M. (1998). The relationship of computer experience and computer self-efficacy to performance in introductory computer literacy courses. *Journal of research on computing in education*, *31*(1), 14-24.

Robins, A. (2010). Learning edge momentum: A new account of outcomes in CS1. *Computer Science Education*, *20*(1), 37-71.

Strayhorn, T. L. (2014). What role does grit play in the academic success of black male collegians at predominantly white institutions?. *Journal of African American Studies*, *18*(1), 1-10.

Watkins, D. (2001). *Correlates of approaches to learning: A cross-cultural meta-analysis.* In R. J. Sternberg & L.-f. Zhang (Eds.), *The educational psychology series. Perspectives on thinking, learning, and cognitive styles* (p. 165–195). Lawrence Erlbaum Associates Publishers.

Wolf, J. R., & Jia, R. (2015). The role of grit in predicting student performance in introductory programming courses: an exploratory study. In the 2015 *SAIS 2015 Proceedings*, *21*. Retrieved online from http://aisel.aisnet.org/sais2015/21

Editor’s Note:

This paper was selected for inclusion in the journal as an EDSIGCON 2020 Distinguished Paper. The acceptance rate is typically 7% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2020.

Appendix A

DVs	Sample 1					Sample 2				
	IDV	Est.	t	p	95% CI L/U	IDV	Est.	t	p	95% CI L/U
Coding SE (C-SE)	SL-CP DL-CP DL-A	0.148 -0.222 0.045	0.935 -1.413 0.279	.354 .164 .781	-0.17/0.47 -0.54/0.09 -0.28/0.37	SL-CP DL-CP DL-A DL-D	0.458 -0.606 -0.026 0.219	2.461 -3.286 -0.120 1.151	.018 .002 .905 .256	0.08/0.83 -0.98/-0.23 -0.46/0.41 -0.16/0.60
	F=.794, p-value=.503, R^2=.046					F=3.448, p-value=.016, R^2 =.243				
Coding Grit (C-G)	SL-CP DL-CP DL-A	-0.048 -0.076 -0.022	-0.512 -0.819 -0.230	.611 .416 .819	-2.84/1.69 -3.26/1.33 -2.58/2.05	SL-CP DL-CP DL-A DL-D	0.193 -0.176 0.061 0.005	2.327 -2.131 0.643 0.067	.025 .039 .524 .947	0.31/4.33 -4.10/-0.11 -1.58/3.05 -1.99/2.12
	F = .657, p-value = .582, R^2 = .038					F = 1.866, p-value=.1338, R^2=.148				

Table 7 Regression Results

Shallow Learning in Computer Programming (SL-CP)

SL1: I try to memorize the steps for solving programming problems presented in the text or in the lecture.
SL2: When I study for the tests I review my class notes and look at solved programming problems.
SL3: When I study for tests I used solved programming problems in my notes or in the book to help me memorize the ‘programming’ steps involved.
SL4: I find reviewing previously solved programming problems to be a good way to study for a test.
SL5: In order for me to understand what technical terms meant, I memorized the textbook definitions.

Deep Learning in Computer Programming (DL-CP)

DL1: When studying, I try to combine different pieces of information from course material in new ways.
DL2: I draw pictures or diagrams to help me solve some programming problems.
DL3: I work on several programming examples of the same type of problems when studying this class so I can understand the problems better.
DL4: I work practice programming problems to check my understanding of new concepts or rules.
DL5: I examine example programming problems that have already been worked to help me figure out how to do similar ‘coding’ problems on my own.
DL6: I classify programming problems into categories before I begin to work them.
DL7: When I work a programming problem, I analyze it to see if there is more than one way to get the right solution.
DL8: While learning new programming concepts, I try to think of practical applications.
DL9: I put together programming ideas or concepts and draw conclusions that were not directly stated in course materials.
DL10: I work on practice programming questions/problems to check my understanding of new concepts or rules.
DL11: When I finish my programming practice questions/problems I check my solution for syntax errors.

<u>Additional Survey Items for Pilot Sample 2</u>

D1: Some programming problems can be visualized using diagrams and models.
D2: I develop models or pictures to help me visualize how programming work.
D3: I model different program modules or functions using some diagramming techniques.
D4: I use some diagramming techniques to understand how programming work.

An Inventory of Privacy Curricula Offerings in Higher Education

Dustin Steinhagen
dustin.steinhagen@trojans.dsu.edu

Chase Lucas
chase.lucas@trojans.dsu.edu

Mary Francis
mary.francis@dsu.edu

Mark Lawrence
mark.lawrence@trojans.dsu.edu

Kevin Streff
kevin.streff@dsu.edu

Dakota State University
Madison, SD 57042, USA

Abstract

Privacy workforce development is a growing need as organizations struggle to find qualified privacy professionals such as Data Protection Officers and privacy engineers. Little has been written about the availability of formal privacy education opportunities that could satisfy this demand. This study inventoried the current state of formal privacy education at institutions of higher education. The inventory included information on 115 privacy programs and 333 privacy courses offered at 99 institutions around the world. Analysis revealed that privacy education is dominated by legal and compliance curricula at the graduate level, with other data privacy opportunities available in smaller quantities.

Keywords: privacy education, privacy workforce development, privacy curricula, privacy in higher education

1. INTRODUCTION

Privacy workforce development is a pressing need in the privacy industry. Half a million organizations have registered Data Protection Officers (DPOs) in Europe since the General Data Protection Regulation (GDPR) went into effect in 2018 (Fennessy, 2019c). The demand for privacy engineers, individuals who understand privacy and can build it into products and services, has grown significantly in recent years (Fennessy, 2019a). Stakeholders who commented on the National Institute of Standards and Technology's (NIST) emerging Privacy Framework identified privacy workforce development as a "critical" need (Fennessy, 2019b). Another source reports that companies from all over the world are "having trouble finding" qualified privacy professionals (Kingson, 2019).

With present and future needs for privacy professionals, it is important to understand the current state of formal privacy education. Security is related to, but different from, the problem of privacy, thus an investigation of privacy workforce development must go beyond looking at security programs in higher education. It is not clear from the literature and existing inventories of privacy programs what privacy opportunities exist in higher education. Existing inventories (International Association of Privacy Professionals, n.d.-b, n.d.-d) are not comprehensive and are not dynamically updated to reflect the current state of available privacy offerings. General information regarding these opportunities is lacking, such as whether opportunities exist for undergraduate students or how many privacy schools exist in each geographic region. Thus, prospective privacy students may not have an effective avenue for identifying institutions of higher education that offer programs within their area of interest.

This paper presents an inventory of 115 privacy programs and 333 privacy courses in higher education from 99 institutions around the world. Previous inventories do not go below the university level when discussing the general state of the privacy education landscape. Several charts are presented, summarizing key statistics of existing programs and courses as well as the universities housing these privacy offerings.

The remainder of this paper is as follows: section 2 summarizes the current knowledge of academic privacy programs generally; section 3 lays out the methodology for inventorying the existing academic programs and courses in privacy; section 4 presents the results of the inventory; section 5 discusses these results; and section 6 concludes the paper and outlines future opportunities.

2. LITERATURE REVIEW

The International Association of Privacy Professionals (IAPP) is the largest non-profit and policy-neutral organization "that helps define, support and improve the privacy profession globally" (International Association of Privacy Professionals, n.d.-a). The IAPP has published two inventories of privacy curricula at institutions of higher education. One is a webpage with information on 74 institutions offering privacy content from around the world, providing details about online availability, whether they can be completed part-time, possible externships/internships and whether prerequisite degrees were required (International Association of Privacy Professionals, n.d.-b). Details about how this inventory was prepared are lacking, and some universities were listed only for having privacy research groups rather than privacy courses. The second IAPP inventory of privacy programs began with a 2019 study of the privacy offerings from law schools that have been accredited by the American Bar Association (International Association of Privacy Professionals, n.d.-d). The study split law schools into four categories: "Tier 1," which were schools that offered a formal concentration or certification in privacy law; "Tier 2," which were schools that offered at least one three-credit course on privacy each year; "Tier 3," which were schools that offered some sort of privacy content, but didn't meet the criteria for being categorized as Tier 2, such as those who offer a one-credit seminar on privacy or have offered privacy courses in the past, but not consistently; and "No Data/Not Counted," representing schools that had no privacy law offerings. The study found that 107 of the 216 (49.5%) law schools had no privacy law content at all, with another 68 (31.5%) schools being categorized as Tier 3 with minimal privacy offerings. The remaining 19% of schools were split between Tier 2 – 30 schools (14%) – with a single course and Tier 1 – 11 schools (5%) – with fleshed out privacy offerings.

Privacy can be split into three major subdisciplines – Legal/Compliance, Management and Technology. Examples of privacy work roles can be found in (Farber, 2018): legal and compliance roles include privacy attorneys and Data Protection Officers (DPOs); privacy managers could include Chief Privacy Officers (CPOs) as well as privacy product managers; and technical privacy roles include privacy engineers and designers of novel privacy-enhancing technologies. This categorization scheme is reflected in the IAPP's entry-level professional certification programs, which split privacy work roles into those of compliance, management and technology (International Association of Privacy Professionals, n.d.-g). In the first half of 2020 ISACA released their Certified Data Privacy Solutions Engineer certification program (ISACA, n.d.). ISACA has dubbed privacy "a growth sector," with their research suggesting that as many as 40% of organizations "lack competent resources" to establish effective privacy programs. In August 2020, business magazine *Inc.* named data privacy firm OneTrust as the fastest growing company in the United States (Hughes, 2020). The inception of the privacy industry in the United States can be traced back to the enactment of the California Consumer Privacy Act in 2018, with the number of privacy vendors more than quintupling from 44 to 259 between 2017 and 2019 (Ingram, 2020).

The IAPP has launched a program called Privacy Pathways in which they help universities to build out their privacy curricula (International Association of Privacy Professionals, n.d.-e). Information about this program is lacking, although it appears that it is primarily focused on bolstering law programs. In France, universities have been moving fast to train DPOs and combat the talent shortage facing that country (Abboud, 2018), with another author suggesting that a time when students will commonly go to school for privacy studies is fast approaching (Hulefeld, 2018). On May 3, 2019, privacy scholar Daniel Solove tweeted the following: "Ridiculous that most law schools don't have a privacy law course let alone a faculty member doing scholarship in the field. It's time for law schools to wake up" (Solove, 2019). Solove suggests that at a minimum law schools should teach a single course in privacy law, but ought to teach several (Solove, 2016). Kevin Streff, founder of SBS Cybersecurity and Professor of Information Assurance at Dakota State University, stated "data privacy education is going to explode in the coming years" (personal communication, June 15, 2020).

Industry training in privacy exists and privacy offerings are widely available on online learning platforms. For instance, searching for "privacy" on LinkedIn Learning (formerly Lynda.com) gives 1,542 total hits, with 64 courses and 1,494 videos (LinkedIn, n.d.). Coursera has 128 results for "privacy," 27 of which are specializations, certificates or degrees (Coursera, n.d.). Pluralsight has 114 courses dealing with privacy topics (Pluralsight, n.d.). Many of these search results likely included a spectrum of privacy content, spanning dedicated courses to results that barely touch on privacy but otherwise contained the word in their metadata - a full analysis was not performed on these results. The recent growth of privacy curricula in higher education is in stark contrast to the corporate world, where some level of privacy training has been commonplace for several years (Solove, 2012). The International Association of Privacy Professionals provides several training classes (International Association of Privacy Professionals, n.d.-f) and online privacy courses (International Association of Privacy Professionals, n.d.-c). Privacy is also being incorporated into information security industry training as well – for example, privacy is a major focus of SANS Institute's "Law of Data Security and Investigations" course (SANS Institute, n.d.).

3. METHODOLOGY

Data on privacy programs, privacy courses and their associated universities were all gathered as part of this inventory. The two existing inventories provided by the IAPP served as the foundation of this inventory, providing 82 universities, 97 programs and 288 courses. Additional programs and courses were found by conducting exploratory searches with Google and DuckDuckGo between late 2018 and early 2020 utilizing keywords such as "privacy degree" and "privacy certificate." Thus, the whole dataset was manually collected through web searches. Additionally, any privacy offerings and universities that were known to the authors were also included. Utilizing these methods, privacy offerings from 17 additional universities were included, for a combined total of 99 universities.

Except for international programs mentioned in the IAPP inventories, only courses and programs that were available for review in English were included in this inventory. Only courses that explicitly mentioned "privacy" or "data protection" in the title of the course or in their course descriptions and had a major focus on one or both topics were included in this inventory. Courses that generally mentioned compliance, ethics, cybersecurity or other concepts involving or related to privacy and data protection were not included in this inventory unless privacy or data protection were a major theme of that course, as suggested by the course title or course description. A program was included only if it contained at least one privacy course, either offered as an elective that counted towards the program or as part of the core curriculum. The program itself did not need to include "privacy" or "data protection" in its title or otherwise have its focus be on privacy. Individual privacy courses that were not clearly associated with a degree program were represented in the courses count but did not contribute to the degree program count. Universities in existing inventories that did not have privacy curricula upon inspection were excluded from this study.

The following summarizes the main data points collected as part of this inventory. All universities, courses and programs were categorized according to the area or areas of privacy their content focused on, split among the areas of Legal/Compliance, Technology, and Management, which were discussed in the literature review. Courses and programs that belonged to more than one subfield of privacy were given the Interdisciplinary label. For a program to be labeled as anything other than

Interdisciplinary, it needed to have more than 66% of its privacy content be in one subfield of privacy – for example, a program consisting of five privacy courses with three of those courses focusing on the intersection of privacy and technology would be categorized as Interdisciplinary, but if instead four of those courses were technical, then the program would be labeled as Technology. An otherwise technical cybersecurity program with one privacy course focusing on legal and compliance topics would be labeled as a Legal/Compliance. The program level – Undergraduate, Graduate, Minor, or Certificate – was also collected. Similar programs were combined if they fell under the same program level and were similar in nature, such as multiple Juris Doctor concentrations and Master of Laws programs that shared curricula, or doctoral programs being combined with their associated master's degrees.

Courses were labeled with a non-Interdisciplinary category unless the course description or course title suggested mostly interdisciplinary content. For instance, one course description stated "we will examine the privacy protections provided by laws and regulations, as well as the way technology can be used to protect privacy" (Carnegie Mellon University, n.d.-a). This course was categorized as Interdisciplinary because both legal and technical privacy topics were covered, and the two disciplines appeared to be given approximately equal weighting. Universities were also labeled according to the subdiscipline of privacy they focused on, with the most-represented discipline among privacy programs offered at the institution determining the label. Geographic information was collected about the physical location of the universities, utilizing the following categories: Asia-Pacific; Canada; European Union; United Kingdom; US, Midwest; US, Northeast; US, South; and US, West. The four geographic regions in the United States correspond to the four regions utilized during that country's decennial census.

4. RESULTS

The final inventory resulted in 99 universities, 115 programs and 333 courses. Refer to the Appendix for a list of the universities. Figure 1 shows the relative distribution of the levels of the 115 collected privacy programs. Most privacy programs are at the graduate level, which includes law programs, master's degrees, and other non-certificate graduate opportunities. Certificate programs, including those at both the undergraduate and graduate levels, made up the second-largest program level category. There were five undergraduate programs with privacy curricula. The only minor in privacy is offered by the University of Amsterdam (University of Amsterdam, n.d.).

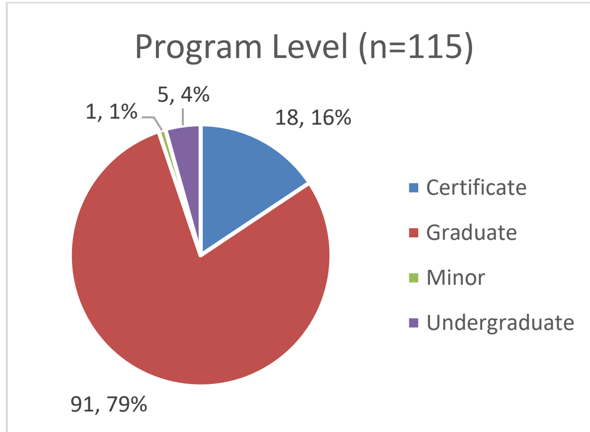

Figure 1: Distribution of Privacy Program Level

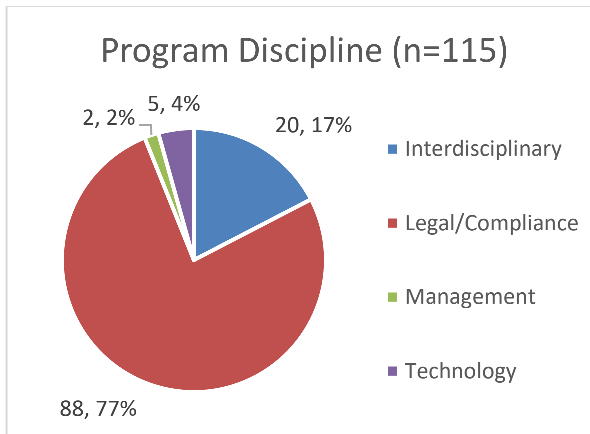

Figure 2: Distribution of Program Discipline

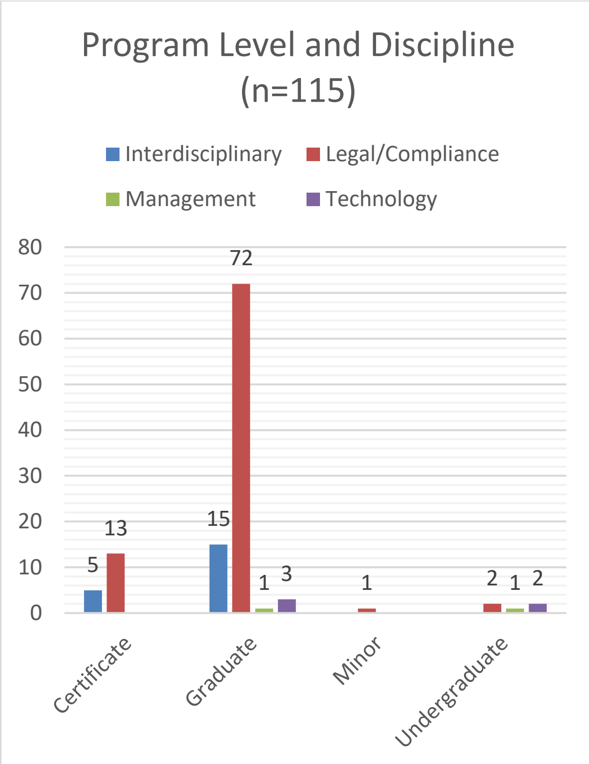

Figure 3: Distribution of Program Level and Discipline

Most privacy programs focus on legal and compliance topics, as indicated in Figure 2. Counting these with Interdisciplinary programs reveals that 94% of the programs inventoried do not primarily emphasize managerial or technical privacy content. Figure 3 shows this program discipline distribution combined with the program level information. Most of the Legal/Compliance education in privacy is at the graduate level or in certificate programs, whereas both Management and Technology are distributed similarly between graduate and undergraduate programs. Interdisciplinary education in privacy was represented at the graduate level and in certificate programs but was completely lacking in undergraduate degrees. As Figure 4 shows, privacy programs tend to have few courses, with 39 (34%) having just one course and 57 (50%) of them having only one or two courses. The distribution is most heavily concentrated towards low course counts, with the number of programs negatively correlated with the number of privacy courses.

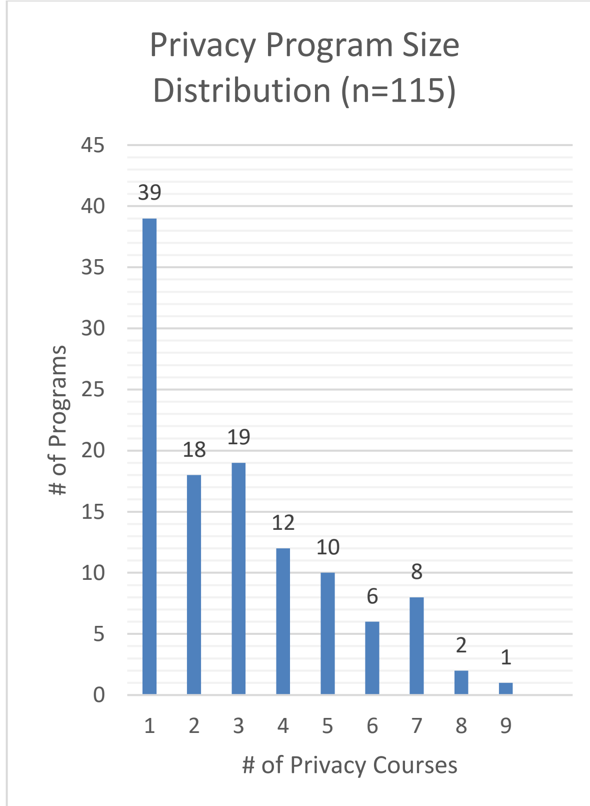

Figure 4: Distribution of Privacy Program Size

The study looked at 333 privacy-focused courses. Figure 5 shows the relative distribution of privacy subdisciplines among the courses. The proportions at the course level are similar as they were at the program level, with one key difference – Management and Technology courses are still in the minority but have approximately double the representation at this level of analysis at 5% and 9% respectively. Legal/Compliance courses made up 76% of privacy courses, with Interdisciplinary privacy courses being slightly more common than technical ones at 10%.

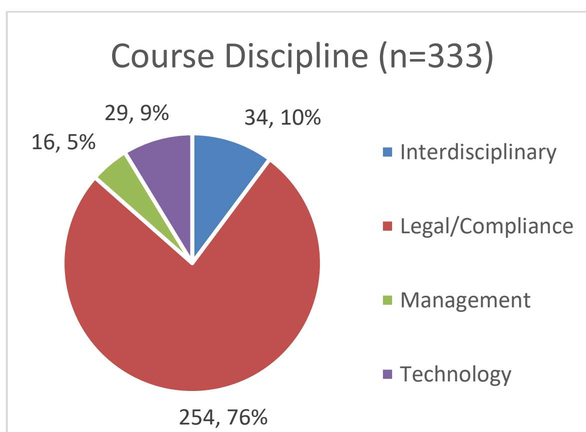

Figure 5: Distribution of Course Discipline

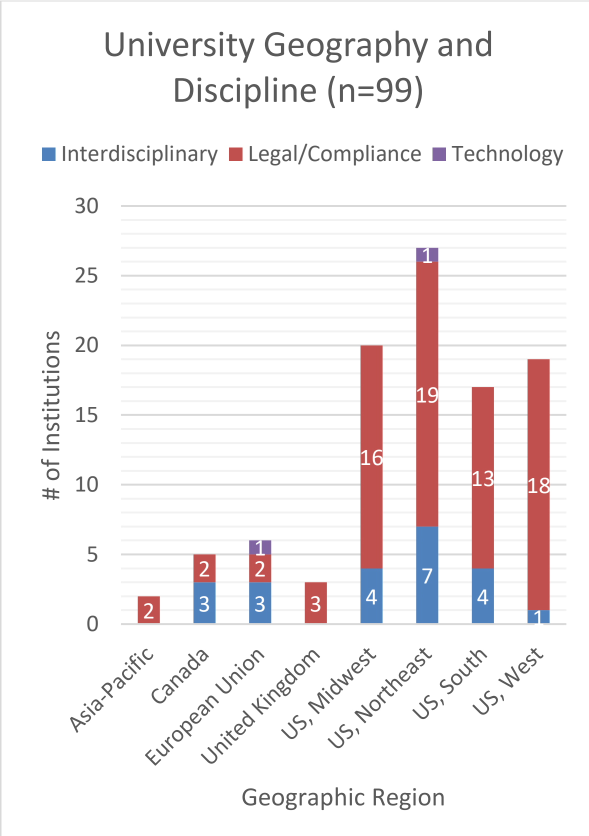

Figure 6: University Locations and Discipline Focus

Figure 6 shows that 83 universities, or 84%, are located within the United States of America. Five institutions are in Canada, nine in Europe and two in the Asia-Pacific region. No university had privacy management as its primary focus, although interdisciplinary programs are available in Europe and North America.

5. DISCUSSION

The data collected reveal several aspects of the current state of privacy education. At all levels of analysis, legal and compliance topics dominate the privacy education landscape, with management and technical offerings scarce or nonexistent. Interdisciplinary courses and programs are more common than management and technical offerings but are far less common than legal and compliance offerings at each level of analysis. These factors combined indicate that the current state of privacy education is narrowly focused on graduate education and legal and compliance topics, despite the current demand for non-legal privacy professionals and privacy being an interdisciplinary issue.

While no Technology or Management programs offered more than one or two courses, a handful of Interdisciplinary and Legal/Compliance programs stood out as exemplary. One example of a mature Interdisciplinary privacy program is Carnegie Mellon University's Master of Science in Information Technology – Privacy Engineering (Carnegie Mellon University, n.d.-b). Their program features six core privacy courses with varying levels of technical and legal/compliance emphasis, a seminar on current topics in privacy, as well as internship and real-world capstone project opportunities. It was the first dedicated university program in privacy engineering and remains the only option for getting a degree in privacy engineering seven years after launching (Fennessy, 2019a). A mature privacy law program is the University of Illinois at Chicago John Marshall Law School's LLM in Privacy & Technology Law program (The Board of Trustees of the University of Illinois, n.d.). This LLM program combines intellectual property topics with privacy, with two of the four required core courses in privacy and six privacy electives to choose from. As per the program's webpage, the program is also "the only graduate law program in the country that emphasizes privacy as part of its core curriculum." Although there were no mature privacy programs that compared to these two examples specifically for Management or Technology, there were several courses in Management and Technology offered as electives as part of other programs, and these two subdisciplines were also frequently covered within Interdisciplinary courses. Thus, there are still opportunities for students wishing to learn about those subfields of privacy.

Privacy programs can be developed in various ways, as indicated by the diversity of program size, degree level and subdisciplines of privacy emphasized among existing curricula. The tendency towards low counts of privacy courses in programs, along with the fact that the inclusion criteria for courses and programs were minimal in this study, is reminiscent of the dawn of cybersecurity education. As part of the early efforts to expand information assurance education, the United States federal government established the Centers of Academic Excellence (CAE) program. To be designated as a CAE back in 2004, a university, in addition to other requirements not related to their information assurance curricular content, needed information assurance to be taught in existing courses, but having dedicated courses on the topic wasn't necessary (National Security Agency, n.d.). Privacy is developing in a similar manner as cybersecurity did, primarily existing as subtopics within courses or as a handful of electives at most universities, as indicated by this inventory. Over time, as privacy matures as a field of study in higher education, programs will expand and become more common. Perhaps some standard-setting body could establish an analogous CAE program for privacy to incentivize excellence and competition in privacy education.

At the university level, most privacy education discovered as part of this inventory is focused on the legal and compliance aspects of privacy, with schools almost evenly spread out across the four regions of the United States. Most of these universities are law schools that offer one or more privacy courses. This means that law students attending schools in the United States who are looking for privacy content have a wide array of options. Dedicated privacy management and technology offerings are too sparse to judge what program maturity in these areas consists of. In general, privacy education is still incredibly rare in the United States of America, with privacy programs being offered at only 83 of the 4042 (Institute of Education Sciences, n.d.) institutions of higher education, or in just over 2% of all American institutions.

Another noteworthy insight is that no university was found to emphasize the managerial subdiscipline of privacy, although several universities had Interdisciplinary privacy curricula that included management curricula and two management programs were found. Additionally, although not as lacking as management options, only two universities specifically addressed the technology subdiscipline of privacy. It is far more difficult for students who are interested in technical or managerial privacy curricula to find appropriate educational opportunities.

6. CONCLUSIONS

Future improvements to this inventory could include seeking out international, non-English programs to get a more global perspective on the state of privacy education. Research institutes, centers and labs with a privacy focus could also be inventoried, which would be valuable for prospective students interested in privacy research opportunities. The data collected for this study could be made more granular. The online availability of courses and programs would help those students working full time or those looking for distance opportunities. Collecting data through direct communications with institutions could help prevent faulty, outdated, or misleading course descriptions from influencing the data. Alumni from privacy programs could be interviewed to document and compare privacy curricula for job alignment and quality. A privacy curricula maturity model for each subdiscipline of privacy could be developed and used to rate current privacy offerings and guide their development.

The inventory presented in this paper could form the basis of a continuing reference database for those interested in professional development in the privacy field. Such a database could be queried for privacy institutions, programs and courses that meet specified criteria, and could be updated as new privacy offerings are made available by schools. This database could be invaluable for all who have a stake in privacy workforce development, such as prospective students, institutions of higher education, career counselors and recruiters of privacy talent graduating from privacy programs.

This study illustrated that much work is necessary before all aspects of privacy education are widely available. Undergraduate offerings in privacy are scarce and the managerial and technical aspects of privacy education have not received as much attention as the legal and compliance aspects of the field. Additionally, resources that communicate information about current privacy education opportunities can ensure that latent privacy talent is able to locate appropriate privacy programs.

7. REFERENCES

Abboud, L. (2018, June 26). *France's top universities aim to respond to DPO needs*. International Association of Privacy Professionals. https://iapp.org/news/a/frances-top-universities-aim-to-respond-to-dpo-needs/

Carnegie Mellon University. (n.d.-a). *17-333 / 17-733 / 19-608 / 95-818 (previous course numbers: 8-533, 17-533, 15-508, 17-801): Privacy Policy, Law, and Technology*. https://cups.cs.cmu.edu/courses/privpolawtech.html

Carnegie Mellon University. (n.d.-b). *Courses*. https://privacy.cs.cmu.edu/plan/courses/index.html

Coursera. (n.d.). *Coursera*. https://www.coursera.org/search?query=privacy

Farber, D. J. (2018, August 10). The advent of privacy engineering. *BigID*. https://bigid.com/blog/the-advent-of-privacy-engineering/

Fennessy, C. (2019a, September 20). *A closer look at Carnegie Mellon's privacy engineering program*. International Association of Privacy Professionals. https://iapp.org/news/a/a-closer-look-at-carnegie-mellons-privacy-engineering-program/

Fennessy, C. (2019b, March 20). *NIST Privacy Framework recognizes critical need for workforce development*. International Association of Privacy Professionals. https://iapp.org/news/a/nist-privacy-framework-recognizes-critical-need-for-workforce-development/

Fennessy, C. (2019c, May 16). *Study: An estimated 500K organizations have registered DPOs across Europe*. International Association of Privacy Professionals. https://iapp.org/news/a/study-an-estimated-500k-organizations-have-registered-dpos-across-europe/

Hughes, T. J. (2020, August 12). *Reflecting on the growth of the privacy industry*. International Association of Privacy Professionals. https://iapp.org/news/a/reflecting-on-the-growth-of-the-privacy-industry/

Hulefeld, M. (2018, April 24). *New certificate in cybersecurity and data privacy law responds to student demand*. International Association of Privacy Professionals. https://iapp.org/news/a/new-certificate-in-cybersecurity-and-data-privacy-law-responds-to-student-demand/

Ingram, D. (2020, February 3). *Can privacy be big business? A wave of startups thinks so.* NBC Universal. https://www.nbcnews.com/tech/security/can-privacy-be-big-business-wave-startups-thinks-so-n1128626

Institute of Education Sciences. (n.d.). *Table 317.10. Degree-granting postsecondary institutions, by control and level of institutions: Selected years, 1949-50 through 2018-19*. U.S. Department of Education. https://nces.ed.gov/programs/digest/d19/tables/dt19_317.10.asp?current=yes

International Association of Privacy Professionals. (n.d.-a). *About the IAPP*. https://iapp.org/about/

International Association of Privacy Professionals. (n.d.-b). *Colleges with privacy curricula*. https://iapp.org/resources/article/colleges-with-privacy-curricula/

International Association of Privacy Professionals. (n.d.-c). *Point. Click. Learn.* https://iapp.org/train/online-training/

International Association of Privacy Professionals. (n.d.-d). *Privacy in US law schools: An IAPP Westin Center report*. https://iapp.org/resources/article/privacy-us-law-schools-westin-center/

International Association of Privacy Professionals. (n.d.-e). *Privacy Pathways*. https://iapp.org/connect/privacy-pathways/

International Association of Privacy Professionals. (n.d.-f). *Privacy Training Classes*. https://iapp.org/train/training-classes/

International Association of Privacy Professionals. (n.d.-g). *The value of certification*. https://iapp.org/certify/

ISACA. (n.d.). *CDPSE overview*. https://www.isaca.org/credentialing/certified-data-privacy-solutions-engineer

ingson, J. A. (2019, November 1). *The global shortage of privacy experts*. Axios Media. https://www.axios.com/privacy-expert-shortage-202eda9b-c8fa-4bd4-b1c1-11844812ab3e.html

LinkedIn. (n.d.). *Learning*. https://www.linkedin.com/learning/search?keywords=privacy

National Security Agency. (n.d.). *Criteria for measurement*. U.S. Department of Defense. web.archive.org/web/20040417215834/http://www.nsa.gov/ia/academia/caeCriteriaList.cfm

Pluralsight. (n.d.). *Pluralsight*. https://www.pluralsight.com/search?q=privacy

SANS Institute. (n.d.). *LEG523: Law of Data Security and Investigations*. https://www.sans.org/course/cybersecurity-law-data-security

Solove, D. (2012, October 1). *Privacy training: An emerging part of the corporate education canon*. International Association of Privacy Professionals. https://iapp.org/news/a/2012-10-01-privacy-training-an-emerging-part-of-the-corporate-education/

Solove, D. (2016, January 15). *Teaching information privacy law*. TeachPrivacy. https://teachprivacy.com/teaching-information-privacy-law/

Solove, D. [@DanielSolove]. (2019, May 3). *Ridiculous that most law schools still don’t* *have a privacy law course let alone a faculty member doing scholarship in* Twitter. https://twitter.com/DanielSolove/status/1124286774145163265

The Board of Trustees of the University of Illinois. (n.d.). *LLM in Privacy & Technology Law*. https://jmls.uic.edu/academics/centers/ip-privacy/llm-it-privacy-law/

University of Amsterdam. (n.d.). *Privacy Studies*. https://www.uva.nl/en/programmes/minors/privacy-studies/privacy-studies.html

Appendix

Table A1: List of the 99 universities included in this inventory

Albany Law School	Robert Morris University
American University Washington College of Law	Rochester Institute of Technology
Baylor University	Ryerson University
Boston College Law School	Saint Louis University School of Law
Boston University Metropolitan College	Santa Clara University School of Law
Boston University School of Law	Seoul National University School of Law
Brooklyn Law School	Seton Hall University School of Law
Brown University	Southwestern Law School
Cardozo Law	Stanford Law School
Carnegie Mellon University	Tilburg University
Chicago-Kent College of Law	Touro Law Center
Cleveland-Marshall College of Law	Universidad San Pablo CEU
Columbia University	University at Buffalo
Dakota State University	University of Alaska Southeast
Davenport University	University of Alberta
DePaul University Law Center	University of Amsterdam
Drexel University	University of Arizona
Duke University	University of California, Berkeley
Embry-Riddle Aeronautical University	University of California, Hastings
Florida State University College of Law	University of California, Irvine
Fordham University School of Law	University of Chicago
Franklin Pierce University	University of Colorado Boulder
George Mason University	University of Denver Law School
George Washington University Law School	University of Florida
Georgetown University	University of Guelph
Georgia State University	University of Illinois School of Information Sciences
Golden Gate University Law School	University of Maine
Harvard University	University of Maryland
Indiana University	University of Massachusetts Amherst
Iowa State University	University of Massachusetts School of Law
John Marshall Law School	University of Minnesota
Johns Hopkins University	University of New Hampshire
Karlstad University	University of North Carolina Chapel Hill
KTH Royal Institute of Technology	University of Pennsylvania
KU Leuven	University of San Diego
Loyola Law School	University of San Francisco
Loyola University of Chicago	University of Southampton
Marquette University	University of Southern California
Mitchell Hamline School of Law	University of Strathclyde
New Jersey Institute of Technology	University of Texas, Austin
New York Law School	University of Texas, El Paso
New York University	University of Toronto
Northeastern University	University of Utah S. J. Quinney College of Law
Northwestern University	University of Washington
Norwich University	Victoria University of Wellington
Ohio State University	Washington University
Ottawa University	Wayne State University Law School
Pepperdine University	Western Michigan University
Purdue University Global	William & Mary Law School
Queen Mary University of London School of Law	

Online teaching effectiveness: A case study of online 4-week classes in a graduate information systems program

Joni K. Adkins
jadkins@nwmissouri.edu

Cindy Tu
cindytu@nwmissouri.edu

School of Computer Science and Information Systems
Northwest Missouri State University
Maryville, MO 64468, USA

Abstract

The COVID-19 pandemic in Spring 2020 caused college classes to be changed from face-to-face classes to online classes. For some students, this was their first introduction to online courses. The pandemic resulted in many summer classes also to be online. Two graduate information systems courses typically taught in face-to-face four week summer classes were changed to online for the summer 2020 session. The courses used both recorded videos for asynchronous instruction and daily class Zoom sessions for synchronous meetings. In addition, breakout rooms, help sessions, group projects, and peer review were also used in these courses. This case study outlines how various aspects of the classes were changed and shares the results of student surveys regarding their experiences.

Keywords: COVID-19, online classes, disruption, asynchronous, synchronous

1. INTRODUCTION

The coronavirus COVID-19 pandemic in 2020 had a profound impact on all aspects of life including higher education. Colleges around the world transitioned to online instruction in an effort to stop the spread of the virus. Most four-year universities in the United States changed to remote emergency teaching. One survey found 93 percent of institutions changed to online instruction in the spring of 2020 (Johnson, Veletsianos, & Seaman, 2020); however, 70 percent of university faculty had not taught a virtual class prior to the pandemic (Hechinger & Lorin, 2020). Researchers and administrators recognized the unusual turn of events and were hesitant to criticize faculty or their teaching during the pandemic (Johnson et al., 2020). Our institution echoed this idea and stressed that faculty needed to be forgiving of themselves and also extend additional consideration and kindness to students during the turbulent time.

Discussions now turn to what is next for higher education. In the fall 2019 term, 15 percent of the total undergraduate population took all classes online (Hechinger & Lorin, 2020), and the percentage will likely be higher in fall 2020. Many see the sudden change to online as an emergency issue (Hodges, Moore, Lockee, Trust, & Bond, 2020). Others view the switch to online for Spring 2020 as a great online experience that will serve as a way to foster better teaching and learning practices (Shinn, 2020). There's little doubt that the pandemic will change higher education practices.

Currently international students make up 100 percent of the students in a M.S. in Information Systems program at a regional state university in the Midwest. International students are generally restricted to enrollment only in face-to-face classes. This requirement was relaxed for international students, allowing them to take online courses in 2020. The COVID-19 situation led this university to change all summer courses online so the two required summer courses, Project Management in Business and Technology and Professionalism in the Information Systems Environment were quickly moved to an online format. Results of student surveys provide insight into what worked well in these online classes. This case study begins with a brief literature review related to COVID-19 and online course development. Then the format and delivery of the two courses are outlined and the results of the surveys shared.

2. LITERATURE REVIEW

Given the disruptive nature of this change to online learning, higher education is experiencing emergency remote teaching, "a use of fully remote teaching solutions for instruction or education that would otherwise be delivered face-to-face or as blended or hybrid courses and that will return to that format once the crisis or emergency has abated" (Hodges et al., 2020, para. 13). The classes that were suddenly moved online should not be compared to well-designed online classes. The typical time to plan and develop an online course is six to nine months prior to teaching the course with the instructor getting comfortable with the online environment in the second or third iteration (Hodges et al., 2020). While many faculty members are accustomed to teaching solo, good online classes are often developed with a team approach, including instructional design specialists (Shinn, 2020).

Changing from face-to-face instruction to online teaching requires the instructor to alter nearly every aspect of teaching. They should not use a standard lecture and notes and deliver it online (Shinn, 2020). Faculty need to be prepared to alter their content and delivery. This may mean incorporating some flipped classroom practices where students review material before a synchronous class session. Faculty with experience with flipped classrooms may have fewer challenges moving online as those without that experience (Shinn, 2020).

Current articles outline some of the practices that faculty used in the emergency remote teaching Spring 2020 semester. A big question for faculty was whether to require synchronous sessions or allow students to work on their own with asynchronous content. Baker, Unni, Kerr-Sims, and Marquis (2020) found that students did not support Zoom sessions as it reduced the flexibility that students wanted. Another student survey of an anatomy class found that 62.2% of the students wanted asynchronous content such as videos uploaded to YouTube (Roy, Ray, Saha, & Ghosal, 2020). Aragon and Wickramasinghe (2016) determined that the number of videos that students watched positively impacted student performance. Scagnoli, Choo, and Tian (2019) also discovered the use of videos positively influenced learning experiences and that graduate students were more likely to watch videos than undergraduate students were. Others recognize that synchronous sessions are sometimes necessary due to the nature of the activity. For example, a group working on a simulation needs to meet online at the same time (Kreie, Johnson, & Lebsock, 2017).

Many used breakout rooms in Zoom as well as online collaboration tools such as Google Docs for student teamwork (Yager, 2020). In a survey of 897 university faculty, 83 percent used their institution's Learning Management System to distribute material, 80 percent used synchronous video tools such as Zoom, Google Hangouts, and GoToMeeting, and 65 percent generated their own content by created videos and allowing students to access on their own time (Johnson et al., 2020). Sixty-four percent stated they changed the assignments or assessments that were previously planned, and 48 percent lowered their course expectations (Johnson et al., 2020).

Clearly, lots of changes were necessary to accommodate the emergency remote teaching required. Many faculty faced the additional challenges of holding student attention in an online environment. It is not surprising that online education deals with attention span issues as students are tempted to multitask while attending online classes (Govindarajan & Srivastava, 2020). Academic integrity also is an issue for a class that is suddenly moved online as teachers need to determine how to protect exams and use tools to deter cheating (Hechinger & Lorin, 2020).

The switch to online learning for students may have provided some positive impacts. Yager (2020) said that students can grow into more independent and self-regulated learners, and quiet students might find their voice through the online environment. In addition, the pandemic allowed some students more time to study and become more reflective in their school work. One of the biggest challenges for students was the availability of dependable, high-speed Internet service. Faculty should be aware of this and find out from their students what kind of technology challenges they may face (Johnson et al., 2020). Disrupted Internet connectivity in India was the largest constraint in the online anatomy class (Roy et al., 2020). Unfortunately, online education tends to amplify the digital divide as some students have computers, devices, broadband connection, and a quiet place to work on online classes while other students lack these necessities (Govindarajan & Srivastava, 2020). Faculty need to be aware of these challenges that students may face when they are not in the classroom.

3. FORMAT FOR PROJECT MANAGEMENT

Spring 2020 courses all finished online. Immediately following spring courses, the summer Project Management 4-week course began. The Project Management course covers the skill set needed to successfully lead an information system development team in effective project management using the constraints of scope, time, cost, and quality. Current behavioral and technical tools of project management were presented within the context of the information systems development process. Some of the work required the use of workgroups and teams.

Before the course started, video lectures for all chapters and software tool tutorials were recorded in VidGrid and posted in the Canvas course site. VidGrid is an external tool available through Canvas that allows for easy screen recording with voice and an option to have machine-translation done for the required closed captioning. The length of videos were between 5 and 30 minutes. The goal was to keep most lectures under 15 minutes so that students would maintain interest when watching the videos. For the longer videos, the chapter function in VidGrid was utilized. Students were able to quickly go to the part of the chapter they were be interested in reviewing.

The class used a team-based learning structure where students were assigned to teams of 4-5 members. The teams remained the same for all projects and discussions. Daily study plans were sent to all 37 students at the beginning of the course. Students were required to review the lecture slides and reading materials and watch the videos before the class meeting. In the daily 40-minute Zoom class meeting, the instructor summarized the knowledge points and answered students' questions. Pop quizzes were randomly given during the daily Zoom sessions to assess the students' study progress; the quizzes were administered through the LMS, Canvas. Then students were grouped into breakout rooms for team discussion and to work on their team project. The Zoom platform has breakout rooms which allow the leader to randomly or deliberately assign students to smaller groups. In the breakout rooms, students can only see the other members in their room. Students in breakout rooms can request to have the instructor join their room. The instructor visited different rooms and joined team discussions.

In addition to the mandatory daily Zoom class meeting, there were two separate Zoom help sessions led by teaching assistants (TAs) every day. It was optional for students to join these sessions. To ask for further assistance, students could send emails to the instructor or TAs. Individual students or teams could also invite the instructor to an additional Zoom meeting to discuss assignments or projects.

Students were expected to complete many group activities. Teams were given daily discussion questions, and they submitted discussion reports right after the Zoom class meeting. There were two group assignments, requiring students to practice different group coordination and communication tools. Teams applied all these tools to their group projects, following five milestone requirements. Team presentations were done via Zoom. Every team recorded its Zoom presentation and submitted to the course Canvas site.

Three exams were given to students for course assessment. Students were required to use Respondus LockDown Browser plus Webcam to take the exams. This tool worked well for remote proctoring.

This online course required students to practice both self-study and teamwork. By applying different online tools, all students successfully completed this course within 4 weeks. Course assessment methods were almost the same as the face-to-face course version. The only difference was the presentation. Students did not have chance to present in front of the whole class.

All presentations were completed via Zoom recordings by teams.

4. FEEDBACK TO PLAN SECOND CLASS

Towards the end of the Project Management course, a survey was sent to all students to get some feedback to help plan the second course, Professionalism. Since Zoom sessions were used in the Project Management course, the students were asked about the number and length of those sessions. Most students (58 percent) thought one session should be required daily while 42 percent thought 2-3 Zoom sessions should be required daily (Monday through Thursday). The majority (61 percent) of the students thought Zoom sessions should last 30-40 minutes each.

The students were also asked an open-ended question about how the Zoom sessions with approximately 40 students were working. Over 87 percent of the students said Zoom sessions with 40 was okay; some cited that the number of students didn't matter while others mentioned that they liked hearing everyone else's questions. Most students did not have concerns ahead of the class. Another open-ended question asked about what was most important for them to have a good learning experience. The most common responses (in order with highest first) were course interaction with students and teacher, recorded videos, ways to get help, details on assignments, and good communication.

5. FORMAT FOR PROFESSIONALISM

Using the data from the survey, the content and delivery for the Professionalism course was moved to online. The course includes IT ethics, job search materials, and business communication. Prior to the course beginning, video lectures were recorded for each of the 10 chapters in the Ethics book. These were done in VidGrid in the Canvas LMS and included a script for closed captioning. Each lecture was about 20 minutes. Students were to watch these on their own. Two quizzes and one exam were given that included questions about ethics. In addition, some class discussion and essay questions also used the ethics material.

In the one required daily Zoom session, the instructor reviewed resumes, cover letters, interviews, and oral and written communication topics. Nearly all students had perfect attendance at these sessions. A graduate assistant was also in the session to help with attendance and keep up with the class material and announcements.

Breakout rooms were used for three activities: peer review of job materials, practice writing business messages, and discussion of ethics situations. During the peer review sessions, the entire class started together in one Zoom session where the requirements were covered. Then students were randomly put in breakout rooms with 4 or 5 students. The instructor visited each breakout room and often reviewed a cover letter or resume so all students in that group could hear feedback. The students shared their screens with each other during peer review. When practicing writing business messages, students were given a situation and had to respond with an email. They wrote their email in their breakout room and then submitted it to Canvas. These were graded and the best ones reviewed in the following day's Zoom session. For the ethics discussions, each breakout room was assigned a situation to discuss. Then the students would return to the main session and share their answers or they would submit their written answers to Canvas. The breakout rooms often took longer than anticipated and the Zoom sessions lasted longer than the planned 40 minutes. Most Zoom sessions lasted approximately 75 minutes.

The etiquette luncheon with a meal and a speaker to discuss the rules and allow the students to practice changed to a Zoom session with a guest speaker on dining etiquette. Mock interviews were also done via Zoom. The students had a pre-interview Zoom session with a graduate assistant right before their mock interview to make sure they had appropriate dress and materials ready and to answer any last minute questions. Then the student met with a business professional for a mock interview.

Students were able to get help with assignments in various ways. They could send emails to the instructor or graduate assistant. They could have individual Zoom sessions with the instructor. Several of these sessions were conducted for resume and cover letter review. There were also two daily Zoom help sessions led by the graduate assistants.

The class presentations were probably the most challenging to adapt to online. Students did two elevator pitches and one ethics presentation. For all three videos, the students had someone else video them giving their presentation and then the file was uploaded to Canvas. Students were graded on presentation skills so they had to be in the video. The ethics presentation required a PowerPoint so the students had to show the PowerPoint on their computer, another monitor, or a TV as they gave the presentation.

While some activities worked similar to the face-to-face class, some suffered in this online format. The peer review of cover letters and resumes was not as effective. Students usually print these documents so others can write on them. The instructor typically moderates this entire session, telling them what to look for and change as they are reviewing each other's documents. In the breakout rooms, they were not able to write on other's files and seemed hesitant to make suggestions. The students were not required to watch the other student presentations so they missed giving and receiving feedback from their peers and incorporating audience interaction as well as the actual experience of giving a presentation to an audience. Also the students missed the chance to practice dining etiquette with an etiquette luncheon.

6. DATA ANALYSIS

Question 5 = Strongly agree 4 = Agree, 3 = Neutral 2 = Disagree 1 = Strongly disagree	PM Mean 1-5 scale n = 37	Pro Mean 1-5 scale n = 27
Like online more than F2F	2.92	3.07
Zoom class meetings were effective	3.84	4.33
Chapter videos helped me understand course content	3.70	4.04
Online Zoom help session was necessary for class	4.08	4.26
It was not difficult to get help from instructor or GA	4.31	4.22
Could always reach out to instructor or GA for help	4.38	4.41
Effective to do group work w/ online collaboration tools	4.11	3.89
Liked using breakout rooms for class discussion	N/A	4.48
Breakout rooms for peer review were helpful	N/A	4.26
Breakout rooms for group or team work were effective	4.26	4.26

Table 1: Mean scores by class

The instructors gave an optional anonymous survey at the end of each 4-week summer course. Thirty-seven students (100 percent response rate) took the Project Management survey and 27 (73 percent response) took the Professionalism survey. Means for the various questions were generated to see which ones were highest. Two of the questions regarding breakout rooms were only included in the Professionalism survey.

The mean scores and scale are listed in Table 1. The first question asked whether students liked this course as online more than face to face. The scores were neutral, indicating they did not have a strong opinion about this. Overall, students in both classes liked the Zoom class meetings, were able to get help from instructor and GA, believed online Zoom help sessions were necessary, and liked using breakout rooms.

Instructors were interested in knowing whether students actually used the videos that were created prior to the beginning of each class. The analytics for the videos were not available so the researchers had to ask the students about their use. The responses are shown in Table 2.

Response	PM	Pro
Watched all videos	38%	22%
Watched most of videos	30%	41%
Watched some videos	13%	29%
Watched a few videos	19%	4%
Did not watch any videos	0%	4%

Table 2: How many videos students watched

The students were also asked how they interacted with the videos. Of the students who watched any videos, Table 3 shows how they watched them.

Response	PM	Pro
Watched parts interested	27%	31%
Randomly skipped	5%	15%
Watched begin to end	68%	54%

Table 3: How students watched videos

The videos in the Project Management course utilized the chapter function, allowing students to quickly get to a certain part of the video. The students were asked about their use and opinion of this feature. The results are in Table 4 and clearly show that the chapter feature should be considered for use in future videos.

Response	PM
Did not know about chapter feature	3%
Did not use chapter feature	13%
Chapter feature was useful	81%
Chapter feature was not useful	3%

Table 4: Chapter feature in videos

Students had the option to share comments about what they liked about class and what they would like to have changed. The answers to these questions were analyzed to determine the most frequent comments.

In the Project Management course, students commented that they liked learning knowledge and gaining practice in handling a project, but many wished the course could have been longer than four weeks. In the Professionalism course, student comments were generally positive with several mentions of learning a lot about professional topics, good class activities, and the improvement in their resume, cover letter, and mock interview skills. There were only a few random comments about things to improve including making sure all students participate in breakout rooms, exams, and providing more feedback.

7. DISCUSSION OF FINDINGS

The data analysis results provide insight into important factors for future classes. First, well-prepared online materials are necessary for students to conduct self-study. Video lectures, daily study plans, and detailed instructions can provide students with comprehensive help.

Second, appropriate online coordination and communication software and tools should be applied to help students complete course activities. Zoom, VidGrid, Respondus Lockdown Browser, Google tools, etc. were successfully used in this case. Using these tools, students completed all coursework with little or no communication or coordination difficulty.

Third, flexible instructional methods can meet the needs of online students. Synchronous Zoom class meetings and asynchronous activities were both conducted, and students responded that they were effective. In addition, the use of breakout rooms was rated positively and should be continued and/or expanded in the future.

Fourth, communication is important for the online course success. Students have less chance to directly meet with the instructor in an online course setting, but they may need more assistance to complete course work. The set of communication methods including synchronous class meetings, help sessions, group discussions, individual meetings, announcements, and emails can help meet communication needs. The results showed that students were satisfied with the various communication opportunities.

While the results of the case study may provide some insight into the effectiveness of various components of our compressed online summer classes, these results cannot be generalized to apply to other institutions and to courses taught in a regular 15-week semester.

8. CONCLUSION

Students in this case study did not demonstrate a strong preference for online or face-to-face classes when asked in the survey. In the open-ended comments, a few said they "always want face-to-face classes" and "in-person classes are always better" so there is likely a preference for face-to-face instruction for at least some of the students. These comments are not surprising as other studies have found students preferring face-to-face classes. Peslak, Kovalchick, Wang, and Kovacs (2018) studied students from three universities and found students preferred the face-to-face course delivery method was over online. Yager (2020) agreed that face-to-face teaching will be favored over online due to the human connection.

The motivation for moving these classes to an online format was due to population health concerns. In one recent study, 80 percent of the students were not in favor of continuing online after the pandemic subsides (Roy et al., 2020). Regardless of the format of the courses in the future, lessons learned from this case study will help in setting up and implementing future courses.

9. REFERENCES

Aragon, R., & Wickramasinghe, I. P. (2016). What has an impact on grades? Instructor-made videos, communication, and timing in An online statistics course. *Journal of Humanistic Mathematics, 6*(2), 84-95. doi:10.5642/jhummath.201602.07

Baker, D. M. A., Unni, R., Kerr-Sims, S., & Marquis, G. (2020). Understanding factors that influence attitude and preference for hybrid course formats. *E-Journal of Business Education & Scholarship of Teaching, 14*(1), 174-188.

Govindarajan, V., & Srivastava, A. (2020). Education: What the shift to virtual learning could mean for the future of higher ed. *Harvard Business Review Digital Articles*, 1-6.

Hechinger, J., & Lorin, J. (2020). Ready or not, colleges go online. *Bloomberg Businessweek*(4650), 12-14.

Hodges, C., Moore, S., Lockee, B., Trust, T., & Bond, A. (2020, March 27). The difference between emergency remote teaching and online learning. *Educause Review.* Retrieved from https://er.educause.edu/articles/2020/3/the-difference-between-emergency-remote-teaching-and-online-learning

Johnson, N., Veletsianos, G., & Seaman, J. (2020). U.S. faculty and administrators' experiences and approaches in the early weeks of the COVID-19 pandemic. *Online Learning, 24*(2), 6-21.

Kreie, J., Johnson, S., & Lebsock, M. (2017). Course design and technology for synchronous interaction in an online course. *Information Systems Education Journal, 15*(5), 60-67.

Peslak, A., Kovalchick, L., Wang, W., & Kovacs, P. (2018). Attitudes toward course delivery: A multi-university study of online, on-ground, and hybrid instruction. *Information Systems Education Journal, 16*(4), 27-33.

Roy, H., Ray, K., Saha, S., & Ghosal, A. K. (2020). A study on students' perceptions for online Zoom-app based flipped class sessions on anatomy organised during the lockdown period of COVID-19 epoch. *Journal of Clinical & Diagnostic Research, 14*(6), 1-4. doi:10.7860/JCDR/2020/44869.13797

Scagnoli, N. I., Choo, J., & Tian, J. (2019). Students' insights on the use of video lectures in online classes. *British Journal of Educational Technology, 50*(1), 399-414. doi:10.1111/bjet.12572

Shinn, S. (2020). Online overnight. *BizEd*, 19(3), 32-37

Yager, K. (2020). Lessons from the online world of teaching and learning. *mETAphor*(2), 16-19.

Editor's Note:

This paper was selected for inclusion in the journal as an EDSIGCON 2020 Meritorious Paper. The acceptance rate is typically 15% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2020.

The Importance of Faculty/Staff Support During Times of Crisis

Kiku Jones
kiku.jones@qu.edu

Bruce Saulnier
bruce.saulnier@qu.edu

Julia Fullick-Jagiela
julia.fullick@qu.edu

Computer Information Systems/Management
Quinnipiac University
Hamden, CT 06518, USA

Lori N.K. Leonard
lori-leonard@utulsa.edu
Accounting and Computer Information Systems
The University of Tulsa
Tulsa, OK 74104, USA

Abstract

The purpose of this study is to gain a deeper understanding of the relationships between workload, faculty/staff support and mental wellbeing of students during a pandemic. Specifically, we are interested in better understanding the moderating effects of faculty/staff support on the negative relationship between workload and mental wellbeing of students. The findings of the study show significance in the conditional effects. At the highest levels of support, faculty/staff support moderates the relationship between workload and mental wellbeing of students. Faculty/Staff should be prepared to provide high levels of support for students during normal times, but also during times of crisis. Universities should look to provide training to help prepare them.

Keywords: Mental Wellbeing, Faculty/Staff Support, Workload, Online Learning

1. INTRODUCTION

COVID-19 caused major disruptions to the spring 2020 semester in colleges and universities across the globe. Due to health concerns, universities moved their classes online and closed their campuses. With the sudden closing of campuses, students were required to find housing elsewhere. Many students returned home, some stayed near campus with roommates, some international students remained on campus due to an inability to return home, and others found accommodations in homes with people other than their families. Students were also required to find ways to continue their studies not only on a different platform, but also in a totally different environment.

Faculty transitioned their courses from on ground to online within a matter of weeks. Some were teaching online for the very first time. Faculty began using technologies they had access to in the past but had never fully utilized. Some faculty were able to navigate the transition virtually seamlessly, while others experienced quite a few bumps along their journey. Some faculty adjusted their syllabi to lighten the load of the semester, while others adjusted the syllabi in a different manner. Some changed group projects to individual assignments, and many required presentations were changed to papers. Faculty tried to determine if they planned to meet their classes synchronously or asynchronously. All these decisions needed to be made quickly to determine what made the most sense for the students, the content, and the remaining course activities.

Many faculties involved in the transition process saw with clarity that many of their students were underserved when schools and colleges moved rapidly to remote instruction. Indeed, many of the most vulnerable students had great difficulty accessing reliable high-speed internet. Other students could not find a quiet place to study, and many more needed to take on greater responsibilities at home to help support their families who needed to navigate through very difficult times. Articles have begun to surface regarding the lack of access to proper technology and Wi-Fi for students off campus. Some of this was stated to be due to too many people in the home or even neighbors competing for Wi-Fi (Day, 2020). Other articles discussed students in low-income environments may be at a disadvantage when transitioning to online learning (Hoover, 2020). These were all issues to consider when determining the content delivery method.

As if it was not painfully clear before, all faculties need to embrace what it means to be an inclusive learning community. As new semesters approach, the faculty need to embrace the reality that good pedagogy is inclusive pedagogy, regardless of whatever mode they find themselves in. Faculty need to recognize that many students are being asked to learn while living through traumatic circumstances and events, conditions that make it virtually impossible to succeed without intentional support and care from the faculty. This means reaching out to students now to ask them what worked and what did not work during the Spring 2020 transition to online classes. Given the students' experiences and their respective realities, faculty must hear their voices regarding what they need to be successful. It means hearing the students' stories and working to bring their voices into the conversation of the classroom in ways that include all voices. It means being a mentor and a voice of support for students when they are faced with the reality of what they are living through.

A survey of college and university presidents found that 91% indicated they were very concerned or somewhat concerned about the mental health of students (Inside Higher Ed & Hanover Research, 2020). However, not much current research is available for colleges and universities to lean on in trying to understand how to improve student experiences in this regard. In order to address the above concern, the student's mental wellbeing is an important factor which need to be studied first. This study will be examining mental wellbeing during a pandemic and a move to virtual instruction and advising and will hypothesize that workload will negatively impact mental wellbeing and this relationship will be moderated by faculty/staff support.

2. LITERATURE REVIEW AND RESEARCH MODEL

Articles are beginning to emerge regarding psychological stress and the workload of healthcare workers as a result of COVID-19 (Breillat & Birtus, 2020; Taylor, 2020; Thompson, 2020). However, COVID-19's effect stretches far beyond healthcare. One area, in particular, that has been drastically altered is higher education. This study examines workload and faculty/staff support effects on a university student's mental wellbeing during the pandemic. The research model is presented in Figure 1.

Workload

Additional and unexpected work added to the students during a time of transition can be stressful. In a study of 209 first-year undergraduate students, teacher-student relationships and sense of purpose were found to impact the perceived workload of the students, and in turn, the perceived workload impacted student engagement (Xenni, Radford, & Shacklock, 2018). It has been also shown that excessive content in University classes can result in a student feeling overloaded (Feldon, 2007), which is even more exaggerated when a pandemic is added to the mix. Smith (2019) examined associations between over 1200 student perceptions of workload and their wellbeing outcomes. The Wellbeing Process Questionnaire was used for the outcomes. The questionnaire groups outcomes in three categories of positive (happiness+life satisfaction+positive affect), negative (anxiety+depression+stress) and cognitive problems. Workload was significant across all the outcomes. It is important to consider workload as higher perceptions of workload can result in greater stress for students and less engagement (Ruohoniemi & Lindblom-Ylänne, 2009).

Faculty/Staff Support

Student satisfaction can often be attributed to a student's experiences with other students (Rowley, 1996). Without face-to-face support of fellow students during a pandemic, students rely even more heavily on faculty. Hammer, Kossek, Bodner & Crain (2013) studied 823 employees and 219 supervisors in an information technology division of a Fortune 500 firm. Using a four-item scale, the researchers asked the employees about their supervisors' help and support of their work and non-work issues/conflicts. They found that employees who rated their supervisors high on the support measurement scale felt they had "more control over their work hours, less obligation to work when they are sick, lower perceived stress, and higher reports of family time adequacy" (Hammer et al., 2013, p. 294).

Wickramasinghe (2012) surveyed 232 software developers who were part of an offshore outsourcing operation. The study found that supervisor support moderates the relationship between work schedule flexibility and job stress. Additionally, supervisor support has been found to have direct and indirect effects on job satisfaction (Charoensukmongkol, Moqbel, & Gutierrez-Wirsching, 2016) and to improve task performance (Afzal, Arshad, Saleem, & Farooq, 2019).

These studies' findings can be adapted to faculty/staff support at universities. It has been shown that students in a "normal" environment are not always aware of all the university support mechanisms available to them (Roberts, Dunworth, & Boldy, 2018). Therefore, faculty/staff must work even harder to ensure that students are aware of the support that is available to them during difficult times. Web-based learning communities and collaborative group assignments help to promote student support in an online class (Fisher & Baird, 2005).

Kirmeyer and Dougherty (1988) studied workload and supervisor support for police radio dispatchers. After each shift, dispatcher perceived workloads, anxiety, and copying mechanisms were assessed. They found higher supervisor support to moderate perceived workload and to help the dispatcher cope better and reduce his/her stress and anxiety.

Mental Wellbeing

Global health points to a student's overall wellbeing (National Center for Chronic Disease Prevention and Health Promotion, 2020). The PROMIS Global mental and physical health items ask questions regarding the participants overall health, quality of life, overall physical health, overall mental health and mood. Mental wellbeing can be examined for many constituents. Wellbeing of employees has been extensively examined. Wellbeing of employees at work can be linked to management leadership and employee trust (Baptiste, 2008). In a study of 19 social workers, workload and workplace expectations were found to impact wellbeing (Shier & Graham, 2013). Additionally, in a study of 64 employees, workload was found to negatively affect the employee's wellbeing, and organizational support was found to moderate the relationship between workload and distress/blood pressure (Ilies, Dimotakis, & De Pater, 2010).

There have also been studies examining the wellbeing of students. However, studies have not examined student wellbeing during a pandemic that forced all classes online. In a study of 594 students from 55 classes, student perceptions of teacher behavior were found to impact student wellbeing (Van Petegem, Aelterman, Van Keer, & Rosseel, 2008). In a study of Australian students, 410 undergraduates were assessed regarding resilience (Turner, Scott-Young, & Holdsworth, 2017). The study found student resilience to be a precursor to student wellbeing. It also found student resilience to be a factor of his/her "experience, university policy and the interactions between the university, work and home environments" (Turner et al., 2017, p. 707). These are all important aspects during a pandemic and online classes.

Given these previous studies and their findings, the current study proposes the following hypotheses:

Hypothesis 1: Workload will negatively impact mental wellbeing.
Hypothesis 2: Faculty/Staff support will moderate the relationship between workload and mental wellbeing.

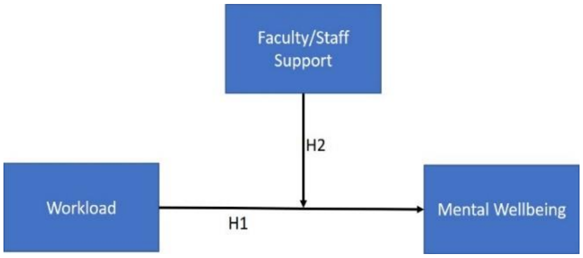

Figure 1. Research Model

2. METHODOLOGY

Students from two US universities were sent emails and asked to anonymously participate in the survey. They were provided with an email link to a survey created in Qualtrics. Some survey questions were adapted from Hammer, Kossek, Bodner & Crain (2013) (Faculty/Staff Supportive Supervisor Behavior Assessment Tool) and PROMIS Global mental health items (Hays, Bjorner, Revicki, Spritzer, & Cella, 2009). Other questions included items regarding environmental factors such as instructor support, personal perceptions, and wellbeing items (Das, 2020). Demographic items were also asked. Items used for each construct can be found in Tables 1-3. Participants were informed that the survey was voluntary and that responses would only be reported in the aggregate.

A total of 127 participants began the survey. Ninety-four completed the survey. Incomplete surveys were excluded. A majority of the participants identified as women (52.6%). Most participants were obtaining a bachelor's or associate's degree (91.6%). Eighty-seven percent of participants were living with family during the pandemic. Most participants were living with three or more people.

3. DATA ANALYSIS AND RESULTS

Harmon's single-factor test was used to determine if common method variance was an issue since several constructs were collected from the same source. The authors entered all variables together. If all variables load on one factor accounting for all of the variance or if one factor accounts for the majority of the variance, common method variance would be present. Using exploratory factor analysis, 3 factors resulted with an Eigenvalue greater than 1.0. The variance explained was between 11.5% and 51%. Therefore, common method variance was not a concern.

Discriminant validity was tested using Spearman's formula (Spearman, 1904). Using a cutoff point of 0.85, all construct pairs were valid, discriminant validity did exist between the constructs.

Construct validity and reliability were tested for all multiple item constructs. Using principal component analysis, factors were extracted. Factors with eigenvalues greater than 1.0 were retained. Varimax rotation was used to indicate high item correlations with a 0.50 cutoff being used.

The items for the Workload can be found in table 1. All of the items of the construct loaded on one factor. The Cronbach's alpha was 0.87. The variation explained percentage was 78.2%. The Workload variable for each subject was calculated as the average of the items.

Workload*
I have too much school work to do.
I have to work extra hard to finish school-related tasks on time.
I have problems with the workload at school
*Scale used: 1 = Never to 4 = Always

Table 1: Workload

The items for the Faculty/Staff Support can be found in table 2. All of the items of the construct loaded on one factor. The Cronbach's alpha was 0.91. The variation explained percentage was 78%. The Faculty/Staff Support variable for each subject was calculated as the average of the items.

Faculty/Staff Support*
Faculty/Staff make you feel comfortable talking to them about your conflicts between school and non-school.
Faculty/Staff work effectively with students to creatively solve conflicts between school and non-school.
Faculty/Staff demonstrate effective behaviors in how to juggle school and non-school issues.
Faculty/Staff organize the work in class to jointly benefit individuals and the entire class.
*Scale used: 1 = Strongly disagree to 5 = Strongly agree

Table 2: Faculty Staff Support

The items for the Mental Wellbeing can be found in table 3. All of the items of the construct loaded on one factor. The Cronbach's alpha was 0.73. The variation explained percentage was 79%. The Mental Wellbeing variable for each subject was calculated as the average of the items.

Mental Wellbeing*
In general, would you say your quality of life is
In general, how would you rate your mental health, including your mood and your ability to think?
*Scale used: 1 = Poor to 5 = Excellent

Table 3: Mental Wellbeing

Means, standard deviations, reliabilities and intercorrelations of the study variables can be found in Table 4 (Appendix A). We used Hayes' (2017) PROCESS macro (Model 1) to test our hypotheses. This macro examines the conditional effects of moderating variables. For our study, we entered workload as the independent variable, faculty/staff support as the moderator, and mental wellbeing as the dependent variable. Table 5 (Appendix A) presents the results.

Support was found for both hypothesis 1 and 2. As can be seen in Table 5, the overall model was significant (p=.000) and had an appropriate R^2 (Chin, 1998, Cohen, 1988; Falk & Miller, 1992).Workload is significantly negatively related to Mental Wellbeing (b = -.18; SE = .08; p = .02). In addition, there is significance in the conditional (moderating) effects. When the Faculty/Staff Support increases, the interaction becomes significant. At one minus the standard deviation, there is no significant effect (b = -0.13, SE = 0.16, p = 0.41). When the Faculty/Staff Support increases to the mean level, there is a significant effect found (b = -0.37, SE = 0.12, p = .003). When the Faculty/Staff Support increases to one plus the standard deviation, there is an even higher significance (b = -0.61, SE = 0.16, p = .000). This suggests that increased levels of faculty/staff support can help students' mental wellbeing when they are finding heavy or difficult workloads, especially during unusual times such as a pandemic in this case. Appendix B provides examples of open-ended responses that further support the need for faculty/staff involvement.

4. DISCUSSION

Support was found for faculty/staff support moderating the effects of workload on mental wellbeing. This indicates that students may in fact utilize faculty/staff support to help alleviate some of the stress and pressure that is felt when workload is perceived to be high. Faculty/Staff can be a valuable resource for students as mentors and advisors. Universities should take the opportunity to provide faculty/staff with the appropriate tools by training them in this area. This training should not only prepare them for the typical semester scenarios, but also for potential crisis mode such as the pandemic. How might faculty/staff be better prepared to serve students in the coming months? What steps should they take now to be ready for students' arrival in the coming semesters? Are there students they haven't heard from who they should be reaching out to now? Today, faculty should be seeking methods to begin interacting with future semester's students. There is still great uncertainty with the coming months, or even a year. Faculty should focus on how their guidance can be used to improve student wellbeing.

5. LIMITATIONS

This study focused on the moderating effects of faculty/staff support on the relationship between workload and mental wellbeing. While we believe this is an important first step in understanding what was happening during the pandemic, we recognize that there are other factors which need to be studied. For example, students' psychological safety and willingness to seek help. Both factors may have impacted a student's mental wellbeing. Future researchers should look at these factors and determine their impact.

In addition, our study looked at two universities. This would be more generalizable if we had a larger sample size. Future researchers should look to replicate the study and gain additional data.

Another limitation of the study is the authors only looked at the mental well-being of the students. During the pandemic, faculty workload was also heavily increased. Just as with students, there may be a negative relationship with workload and mental well-being of faculty. Future researchers should study this and determine if there are any reciprocal effects on the relationships to the student.

6. CONCLUSIONS

This study provides valuable insight regarding faculty/staff support. When given at the highest levels, faculty/staff support moderates the negative relationship between workload and mental wellbeing. This shows the importance of faculty/staff support during a time of crisis, such as the pandemic. Universities should strive to train faculty/staff on how to mentor and advise students, so they are prepared to serve the students appropriately. Future researchers should look for other variables, such as psychological safety, which may impact student mental wellbeing.

7. REFERENCES

Afzal, S., Arshad, M., Saleem, S., & Farooq, O. (2019). The impact of perceived supervisor support on employees' turnover intention and task performance. *Journal of Management Development*, *38*(5), 369-382.

Charoensukmongkol, P., Moqbel, M., & Gutierrez-Wirsching, S. (2016). The role of co-worker and supervisr support on job burnout and job satisfaction. *Journal of Advances in Management Research*, *13*(1), 4-22.

Chin, W. W. (1998). The partial least squares approach to structural equation modeling. In: G. A. Marcoulides (Ed.), *Modern Methods for Business Research* Mahwah, NJ: Lawrence Erlbaum Associates.

Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Lawrence Erlbaum Associates, New Jersey.

Das, J. (2020). COVID-19 student Impact Study. Retrieved April 18, 2020 from https://osf.io/64afc/

Day, R. (2020). The Quarantine is Stressing Our Wi-Fi. EE Times. Retrieved April 18, 2020 from https://www.eetimes.com/the-quarantine-is-stressing-our-wi-fi/#

Falk, R. R., & Miller, N. B. (1992). A primer for soft modeling. University of Akron Press, Ohio.

Feldon, D. F. (2007). Cognitive load and classroom teaching: The double-edged sword of automaticity. *Educational Psychologist*, *42*(3), 123-137.

Fisher, M., & Baird, D. E. (2005). Online learning design that fosters student support, self-regulation, and retention. *Campus-wide information systems*, *22*(2), 88-107.

Hammer, L. B., Ernst Kossek, E., Bodner, T., & Crain, T. (2013). Measurement development and validation of the Family Supportive Supervisor Behavior Short-Form (FSSB-SF). *Journal of Occupational Health Psychology*, *18*(3), 285.

Hayes, A. F. (2017). Introduction to Mediation, Moderation, and Conditional Process Analysis (Second Edition: A Regression-Based Approach ed.). Guilford Publications, New York.

Hays, R. D., Bjorner, J. B., Revicki, D. A., Spritzer, K. L., & Cella, D. (2009). Development of physical and mental health summary scores from the patient-reported outcomes measurement information system (PROMIS) global items. *Quality of Life Research*, *18*(7), 873-880.

Hoover, E. (2020, April 24). Distanced Learning. *The Chronicle of Higher Ed.* Retrieved from June 4, 2020 from https://www.chronicle.com/article/Low-Income-Students-Need/248634

Ilies, R., Dimotakis, N., & De Pater, I. E. (2010). Psychological and physiological reactions to high workloads: implications for well-being. *Personnel Psychology*, *63*(2), 407-436.

Inside Higher Ed and Hanover Research (2020). Responding to the COVID-19 Crisis, Part II: A New Survey of College and University Presidents. Retrieved June 5, 2020 from https://www.insidehighered.com/booklet/responding-covid-19-crisis-part-ii-new-survey-college-and-university-presidents

Kirmeyer, S. L., & Dougherty, T. W. (1988). Work load, tension, and coping: Moderating effects of supervisor support. *Personnel Psychology*, *41*(1), 125-139.

National Center for Chronic Disease Prevention and Health Promotion, (2020). Well-Being Concepts. Retrieved June 5, 2020 from https://www.cdc.gov/hrqol/wellbeing.htm

Roberts, P. A., Dunworth, K., & Boldy, D. (2018). Towards a reframing of student support: a case study approach. *Higher Education*, *75*(1), 19-33.

Rowley, J. E. (1996). Customer compatibility management: an alternative perspective on student-to-student support in higher education. *International Journal of Educational Management*, *10*(4), 15-20.

Ruohoniemi, M., & Lindblom-Ylänne, S. (2009). Students' experiences concerning course workload and factors enhancing and impeding their learning–a useful resource for quality enhancement in teaching and curriculum planning. *International Journal for Academic Development*, *14*(1), 69-81.

Shier, M. L., & Graham, J. R. (2013). Organizations and social worker well-being: The intra-organizational context of practice and its impact on a practitioner's subjective well-being. *Journal of Health and Human Services Administration*, *36*(1), 61-105.

Smith, A. P. (2019). Student Workload, Wellbeing and Academic Attainment. In L. Longo & M. Leva (eds). *Human Mental Workload: Models* *and Applications.* Cham: Springer International Publishing.

Spearman, C. (1904). The proof and measurement of association between two things. *The American journal of psychology*, *15*(1), 72-101.

Taylor, L. (2020). Psychological stress of COVID-19 frontline healthcare workers: rational safety behaviors, evidence-based self-help interventions, and remote mental health support. *Psychosociological Issues in Human Resource Management*, *8*(1), 61-66.

Thompson, D. (2020). Psychological trauma symptoms and mental conditions of medical staff during the COVID-19 pandemic: severe stress, elevated anxiety, and clinically significant depression. *Psychosociological Issues in Human Resource Management*, *8*(1), 25-30.

Turner, M., Scott-Young, C. M., & Holdsworth, S. (2017). Promoting wellbeing at university: the role of resilience for students of the built environment. *Construction Management and Economics*, *35*(11-12), 707-718.

Van Petegem, K., Aelterman, A., Van Keer, H., & Rosseel, Y. (2008). The influence of student characteristics and interpersonal teacher behaviour in the classroom on student's wellbeing. *Social Indicators Research*, *85*(2), 279-291.

Wickramasinghe, V. (2012). Supervisor support as a moderator between work schedule flexibility and job stress. *International Journal of Workplace Health Management*, *5*(1), 44-55.

Xenni, M. J., Radford, K., & Shacklock, K. (2018). Student engagement in academic activities: a social support perspective. *Higher Education*, *75*(4), 589-605.

Appendix A
Data Analysis Tables

	M	*SD*	1	2	3
1. Work Load	2.36	0.80	(.87)		
2. Faculty/Staff Support	5.18	1.32	-.35**	(.91)	
3. Mental Wellbeing	3.22	1.05	-.41**	.47**	(.73)

Cronbach's alpha are found on the diagonals. *p <.05, ** p <.01

Table 4: Variable Statistics

				95% Confidence Level	
DV: Mental Wellbeing	*b*	SE	*p*	Lower	*Upper*
Workload (Direct Effect)	-0.18	0.08	0.02*	-0.33	-0.03
Conditional Effects:					
Faculty/Staff Support 3.86 (-1 SD)	-0.13	0.16	0.41	-0.44	0.18
Faculty/Staff Support 5.18 (SD)	-0.37	0.12	0.003**	-0.93	-0.29
Faculty/Staff Support 6.51 (+1 SD)	-0.61	0.16	0.000***	-0.93	-0.29

Note: Faculty/Staff Support in the conditional table is the mean and +/- SD (standard deviation) from the mean; *p< .05; **p < .01; ***p < .001; Overall model: p= .000***; R^2 = .34

Table 5: Results

Appendix B
Sample of Open-Ended Response

1. *Faculty were very good at working with us to adjust deadlines and workload, eliminating some of the nice-to-have small tasks while maintaining the core workload and helping me learn the concepts I enrolled to learn.*
2. *The only thing that has been difficult moving online that is notable is the group work. I wish that if we had designated group time it was on Zoom in breakout rooms because it is incredibly difficult to hold people accountable. My professors have been super helpful when things go awry, but I wish that there was a way to hold everyone more accountable.*
3. *I felt that it was hard for some of my classes to be online because it requires the professor to know how to utilize technology. I have been getting a lot of busy work and unbeneficial work during online classes. I do not feel productive about this.*
4. *My professors at [university] have made the online transition seamless; very upfront, communicative and understanding.*
5. *The School of Business professors have been by far the best at keeping in touch with their students during this transition. All of them have been incredibly supportive, flexible, and understanding to those who are struggling or maybe need some space. My professors have given me extra time to complete assignments when I inform them of my work schedule that was vamped up due to COVID19 - I have absolute confidence I will pass my courses with all A's while balancing work and family life. My professors want me to succeed, and oftentimes, will check in on me at random to see how life is going and how my job search is coming. The support has been unreal and incredibly appreciated.*
6. *As someone who struggles with anxiety, the recent changes have been quite overwhelming. I have never really struggled much with keeping up with work, but since moving online, I have struggled very much to keep track of deadlines and to do the work to the best of my ability.*
7. *Certain professors are very very understanding of the increased workload, while others continue to pile it on. In one particular class, it is almost impossible to understand the assignments and the adjunct professor does not know how to give personal assistance and is not available. It is also very difficult to show up for zoom meetings at the time of the meeting due to family needs and personal needs daily during this…*
8. *Online learning is surprisingly difficult. I can't find enough motivation to do school work and no peers to ask questions.*

From Engagement to Empowerment: Project-Based Learning in Python Coding Courses

Mark Frydenberg
Computer Information Systems Department
Bentley University
Waltham, Massachusetts
mfrydenberg@bentley.edu

Kevin Mentzer
Bryant University
Information Systems and Analytics Department
Smithfield, Rhode Island
kmentzer@bryant.edu

Abstract

Project-based learning (PBL) engages students deeply with course concepts and empowers them to drive their own learning through the development of solutions to real-world challenges. By taking ownership of and completing a project that they designed, students develop and demonstrate creativity, critical thinking, and collaboration skills. This paper describes two different software development projects, designed with a PBL approach, in Python coding courses at two business universities in the United States, in which students queried real-world data to answer their own questions and interpret the results. The authors contend that projects based on a PBL approach motivate students for self-exploration and allow for the measure of student learning. The authors present their respective projects, share examples of student work, and offer suggestions and lessons learned from implementing PBL assignments in their classrooms. Finally, the authors reflect, through sharing student comments, on how key aspects of PBL are manifest in this project and discuss challenges in offering and managing PBL assignments. With Python's popularity on the rise, these two class examples serve as a model for how instructors can incorporate autonomy in PBL assignments, offering a valuable learning opportunity for students to create software applications that meaningfully demonstrate their coding skills.

Keywords: project-based learning, Python, data analytics, data science, data visualization, coding

1. INTRODUCTION

Project-based learning (PBL) describes a learning scenario where students are engaged developing solutions to real-world problems often of their own design. The process of identifying a problem and developing a solution contributes to learning. Instructors need to specify required tasks, encourage students to think creatively, keep them motivated.

With its foundations in constructivism, which encourages students to learn through designing their own learning experiences, PBL requires a motivating problem or question for students to investigate. This culminates in the students creating original artefacts that illustrate their findings and demonstrate their understanding of a problem (Blumenfeld, Soloway, Marx, Krajcik, Guzdial, & Palincsar, 1991) process of completing such a project moves students from a place of engagement to a place of empowerment as they take control over their own learning, assess their own knowledge and skills, and demonstrate their competencies in a relevant project of their own design.

This paper describes how a PBL approach informed two software development projects given in Python coding courses at two business universities in the United States. The authors present their respective projects and requirements, share examples of student work, provide student reflections, and offer suggestions and lessons learned from implementing PBL assignments in their classrooms.

A contribution of this work is that it illustrates how carefully crafted coding projects such as these can influence student learning. While the literature has addressed PBL approaches in coding courses, this paper has the unique focus of using data analytics tools in a Python coding course to engage students in interacting in a project of their own choosing, and empower them to discern meaning from information by identifying their own requirements for analyzing real-world data.

These research questions guided this study:

- How can instructors design a course assignment that exemplify key aspects of PBL?
- Can a PBL approach motivate students and serve as an authentic measure of student learning?

2. PROJECT BASED LEARNING IN CODING COURSES

Many introductory programming courses include coding assignments of varying complexity, where the instructor specifies requirements or outcomes for students to complete. Assignments often are associated with textbook chapters or learning modules: when the week's lesson covers loops and if statements, the instructor's carefully constructed assignment ensures their use in the solution. All students work on essentially the same assignment (though some instructors may modify an assignment's requirements from semester to semester or within multiple sections of a course, to offer variety and promote academic integrity). In a PBL approach, students create their own questions, focusing on process over product, as "engaging students in the process of inquiry involves guiding them to ask meaningful questions to investigate compelling real-world problems. Through this process, students build crucial problem-solving skills and learn how to generate creative solutions to complex problems" (McKay, Frank, 2017).

Project Based Learning emphasizes student involvement through direct experience in directing their own learning. Ownership of the project is emphasized throughout the project by having the student in control of the project definition. Students utilize creativity through both the unique definition of the project as well as the election of techniques used to execute the project. Collaboration happens when student interact and provide feedback between peers. Finally, critical thinking enables problem solving throughout the project. Figure 1 summarizes these key aspects of PBL.

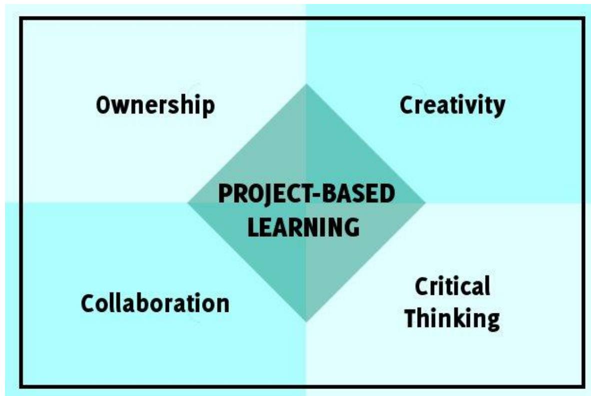

Figure 1. Key aspects of project-based learning [Adapted from (Stefanou, Stolk, Prince, Chen, & Lord, 2013)]

In a well-designed PBL experience, the student has ownership of the project. Student learning outcomes are improved if the project demands both creativity and critical thinking (Rice & Shannon, 2016; Sharkey & Weimer, 2003)(Rice & Shannon, 2016; Sharkey & Weimer, 2003). Finally, in many learner-centered environments, different forms of collaboration, such as learning from and with peers, often improve the quality of course projects (Aditomo, Goodyear, Bliuc, & Ellis, 2013; Jackson & Bruegmann, 2009; Stefanou, Stolk, Prince, Chen, & Lord, 2013).

VanDeGrift describes a learning scenario where students take ownership by creating their own programming problems in an introductory CS 1 course. "Every assignment includes open ended elements to encourage students to decide how to define part of the specification and provide latitude for students to be creative in their design and implementation" (VanDeGrift, 2015, p. 54). Students build their own interpretations of the material based on their own experiences, resulting in projects that foster creativity, maintain interest, and encourage students to take ownership of their projects.

When implementing a PBL scenario in a coding course, assignments are usually of a larger scale, and require students to select the programming constructs, modules, and data analysis most appropriate to implementing or discovering a solution. "Project-based learning, unlike the traditional textbook/lecture approach, motivates the student to do additional work, illustrates to the student the value of the material covered, and most importantly, provides practical experiences that enrich the student's academic growth" (Baugh, 2011, p. 15).

Courses offering PBL differ from those offering individual or group active learning problem-solving exercises. While students often work on specific well-defined problems during class in flipped classroom environments, (Bergmann & Sams, 2014; Whittington, 2004), in a PBL environment, students identify a problem, often open-ended, to investigate, and then implement their solution in a software application. "Project work … requires the student to develop an entire system - a complicated and new task for most students"(Scherz & Polak, 1999, p. 88).

PBL increases student engagement by having students apply their knowledge as they complete learning activities to challenge their understanding and involve them in the learning process, rather than passively watching, listening, or reading about the topic. Projects are adaptable to a student's interests, abilities, and needs. PBL enriches the classroom experience as students work on different problems in assignments of varying durations, requiring them to integrate their knowledge of several topics. The instructor's role shifts from providing solutions to helping students overcome immediate challenges and roadblocks so they can move on independently with their work. Students often work with or share their work with each other.

As students long for finding relevance and autonomy in the classroom, instructors are evolving the way they offer students assignments to demonstrate their knowledge. In a PBL environment, course projects shift from instructors developing homework problems or exams for students to complete, to students identifying their own problems to solve that meet specified learning objectives. Assignments range from defining their own problems to creating their own final exam questions (Brink, Capps, & Sutko, 2004; Brown, 1991; Jones, Jennifer, 2016). This expands the student's role from learner to assessor, as the process of making up one's own project or exam requires determining relevant topics, examining one's own learning and capabilities, and developing a mechanism to demonstrate competency and knowledge. The process requires use of higher order thinking skills (Bloom, 1956) to generate problems that required more than mere memorization or recall of facts.

3. PYTHON COURSE DESCRIPTIONS

This paper describes two different PBL learning assignments implemented in undergraduate Python coding courses at two universities. Students in both classes completed a project in which they had to use real world data to answer their own questions to demonstrate their mastery of several learning outcomes. Section 5 summarizes comments and responses to open-ended survey questions from students as they reflected on their learning and the value of a PBL methodology in completing their projects.

Both courses met in person at their respective universities during the spring 2020 semester until spring break, and then moved to online delivery in March 2020 because of the COVID-19 pandemic. The mid-semester shift online informed the creation of PBL assignments in these classes as both instructors considered alternative means for students to demonstrate their learning outcomes from the course in a way that genuinely reflected their newfound skills. Administering online exams brought many practical concerns; giving students the opportunity to design, build, present, and explain their solutions offered a practical way to evaluate a student's ability to master and apply course concepts.

The next sections describe the two courses in which the authors implemented PBL final projects in lieu of a more standard final exam, such as multiple choice or pencil-and-paper problems.

CS 299: Problem Solving with Coding in Python

CS 299, Problem Solving with Coding in Python, is an experimental elective open to all students at Bentley University, a northeastern U.S. business university. This course introduces problem solving using programming and teaches the fundamental concepts of algorithm development along with the underlying abstractions that are the basis of software systems. Students develop and integrate critical thinking skills by creating solutions to problems in a systematic, algorithmic manner using the Python programming language. In addition to teaching fundamental Python coding concepts, four class sessions included computational thinking topics and methods:

filtering data based on what is relevant (abstraction), developing algorithms, breaking problems into smaller problems (decomposition), and recognizing patterns (Astrachan, Hambrusch, Peckham, & Settle, 2009; Bell & Lodi, 2019; Rich & Hodges, 2017; Sengupta, Dickes, & Farris, 2018). These learning experiences are paramount in developing computational thinking, an ability to solve complex problems from authentic contexts and everyday life situations by decomposing them into smaller steps that are systematic and suitable for automation.

Students completed many small-group coding exercises and commented on each other's solutions during class so their peers could see alternative solutions to the same problems.

Throughout the course, understanding of coding concepts reinforced throughout the course by the development of several standalone applications, in which the instructor emphasizes the importance of writing efficient, clear, and well-structured code. No prior knowledge of Python or other programming languages is required.

This course met for two 80-minute sessions each week in a 14-week semester. The course had 27 students enrolled, 61% of whom had no prior coding experience. Students were primarily a mix of sophomores and juniors, most of whom were Computer Information Systems (CIS) or Finance majors, or CIS or Data Technologies minors. Each class session included instructor-led presentations and demonstrations, and several in-class exercises, completed in small groups, that reinforced the topics presented.

This course presents basic programming concepts and techniques using version 3 of the Python programming language, such as loops and selection statements; data structures (e.g., lists and dictionaries); classes, and objects. Instructors omitted advanced topics such as higher order functions (e.g., map, reduce, filter, lambda), and other topics frequently taught in Java programming courses (e.g., graphics and user interface design), teaching instead, basic capabilities of several popular Python libraries for data analysis: NumPy, Matplotlib, and Pandas. The course also introduced Streamlit (Treuille, Teixeira, & Kelly, 2020), an open-source app framework to code interactive web pages, to display their results. Incorporating Streamlit moves Python applications out of the console window and into a browser, using a simple platform to create web applications and share their work more widely

Several assessments contribute to evaluating a student's performance: five programming assignments (40%), class participation including completing in-class exercises (5%), short practice programs started during and often completed after each class (10%), a hands-on midterm exam (20%), and a design-your-own final project (25%) in lieu of a standard final exam.

Table 1 in Appendix 1 presents the topics covered in the five programming assignments.

ISA 330: Programing for Data Science

ISA 330, Programming for Data Science, is the second course in Python for students majoring in Data Science at Bryant University in the northeastern United States. This course, which has an introductory Python course as a prerequisite, is an advanced Python programming course focusing on common programming tools used for Data Science application development with an emphasis on libraries commonly used by data scientists (such as NumPy, Pandas, Matplotlib). Data analysts often implement their solutions using programming languages such as R and Python. Because of this, the data analyst/scientist must be comfortable in such development environments and be able to understand when a solution needs to be programmatically developed. The course covers hands-on programming techniques for analytics, including web scraping and other data extraction techniques, data transformation, data staging, data analysis, and finally data presentation and visualization. The course gives the students the skills to highlight their capability of producing notebooks appropriate for a data analytics/data science application.

This course runs each semester with one section offered. The students are primarily a mix of sophomores and juniors. Roughly, 75% of the students are data science majors and the rest is a mix of other business or mathematics majors. Due to the heavy hands-on programming aspect of the course, the class has a maximum of 25 students. The course typically meets three times a week for 50 minutes each session.

Even prior to the moving online after spring break, the course had a flipped component where students watched pre-recorded videos of lectures on their own schedule outside of class. This allowed the class time clear up anything that the students were still unsure about and work on in-class exercises meant to reinforce the concepts learned in the recorded lectures.

In addition to the recorded lectures, students worked with provided Jupyter notebooks that demonstrated the topics for the week. As part of their homework, the student had to modify these notebooks to expand, or modify, the notebook's functionality.

Multiple methods of assessment contributed to evaluating student performance including seven programming assignments (30%), three in class hands-on exams (30%), class participation including attendance and quizzes (10%), and the final project (30%).

Table 2 in Appendix 1 presents the topics covered in the seven programming assignments.

4. PROJECT DESCRIPTIONS

Introducing project-based learning assignments in these courses allows students to demonstrate their skills in applying course concepts to solve real-world problems. The variation among student projects and solutions encourages creativity and engagement as students identify the project components that they will implement to meet the project requirements.

CS 299 Project

CS 299 presented several ideas related to computational thinking and good practices for visualizing data in addition to introducing fundamental coding concepts and principles. The final project for the course required students to demonstrate mastery of these concepts. Given a data file containing approximately 3,400 actual Boston-area AirBnB listings available from http://insideairbnb.com/get-the-data.html, the project had students describe two questions for which they would find like to find answers from the data, and design two visualizations (charts, graphs, or maps) to display the results. Appendix 2, Figures 1 and 2 show screenshots of two sample student projects. Students completed the project in these phases:

Phase 1. Design. Describe two questions and two visualizations that you can create to analyze this data. Examples include: What are the most expensive rentals in each neighborhood? Is there a correlation between reviews and nightly prices? How many rentals are available in each neighborhood? Describe how your queries will be interactive using Streamlit user interface elements. (Time allowed: 4 days.)

Phase 2. Build. Build the solution in a well-documented and structured Python program. (Time allowed: 1 week)

Phase 3. Present and Review. Create a five-minute video (if attending class synchronously is impossible) or present in class. Students watched each other's presentations, and evaluated them using an online form, based on perceived complexity (compared to their own projects), the student's ability to explain their code, what they liked the best about the project, and suggestions for improvement. (Due with project submission.) The instructor needed to approve all proposals before the implementation phase, to ensure they were of adequate complexity.)

Incorporating Streamlit widgets enabled students to create a user interface enabling interactive queries. For example, a user might interact with a slider to specify a maximum rental price and a dropdown list to select a neighborhood. The display shows on a map all homes in that neighborhood whose rental price is below the specified price. As the slider updates, the results update automatically, as shown in Appendix 1, Figures 1 and 2.

If presented in class, students reviewed the presentations of their peers. Involving students in the direct assessment of their classmates' project required students to compare the quality and complexity of their solutions with those of their peers. Asking students to provide praise and constructive recommendations placed them in the role of being active listeners, and the quality of the feedback they provided in written comments to their classmates contributed to their overall project grades.

Students compared their solutions with those of their peers and noting innovations such as, "The Map showing the available listings connected to the user's input was very nice." A student commented that they liked how a classmate included photos of homes in different neighborhoods, "which gave the app a more visual appeal." "It makes it seem like a real website." Students also offered constructive suggestions for improvement, suggesting, "Maybe you could connect the histogram with the data that you filter at the beginning. That way [we] could see the range of prices for each neighborhood" and "The maximum price slider only goes up to 499 but there are listings that are left out because they are more expensive than that."

ISA 330 Project

The "Twitter Project" is an individual project that teaches students how to interface with the Twitter

APIs and explore a dataset of their own choosing. Appendix 3, Figures 1 and 2 show screenshots of two sample student projects. Students completed the project in these phases:

Phase 1: Prepare. Students set up their own Twitter developer accounts. The class explored the Twitter APIs and discussed pulling historical Tweets versus setting up a Twitter listener. A sample notebook was shared that allowed students to "listen" for Tweets immediately after receiving their credentials from Twitter. By choosing a common hashtag or Twitter handle, students were able to see their listener program working before they exited the classroom. We used the handle @RealDonaldTrump due to the high volume of Tweets posted at that time referencing this handle.

Phase 2: Explore. The goal of the project was to compare Tweets on two different topics on Twitter. For example, they could choose "Nike" versus "Adidas", "Microsoft" versus "Apple", "Red Sox" versus "Yankees". They had two weeks to collect real time data using the Twitter streamer API. Once the student had an idea of a topic, they then used the Twitter website to explore the data. This helped the student confirm that the data they retrieved matched what they were interested in studying. In some cases, the students found their topic was too broad, while in other cases it was too narrow. Students refined their selection of handles and/or hashtags to get a dataset that represented the topic they wished to explore.

Phase 3. Acquire Data. Once the student had their targeted list of handles and/or hashtags, they set up a Twitter Listener by modifying the sample Jupyter Notebook. Students collected data and stored it locally on their laptops.

Phase 4: Analyze Data. During several weeks, students explored and shared with each other through informal class presentations, different ways to analyze the dataset. Students began their analysis performing a sentiment analysis on the Tweets making use of code shared with them to assist with this task. Students were encouraged to find examples of other techniques and to walk the whole class through the implementation of that technique. For example, one student shared the implementation of a word cloud using the text of the Tweets.

Phase 5: Reflect and Summarize. The final deliverable included an executive summary of their analysis along with their Jupyter Notebook.

5. STUDENT REFLECTIONS

While the PBL assignments in their respective courses were different, student reflections from both courses suggest that students shared common experiences while completing them. The four key aspects of PBL (ownership, collaboration, creativity, and critical thinking) (Stefanou et al., 2013) provide a reference for evaluating student reflections on their experiences completing these projects.

Ownership

Students in both courses commented on the value that a PBL assignment offered them to demonstrate the competencies and skills they learned in the course, which often exceeded their own expectations.

From CS 299 students:

"I liked that this final provided us with a concrete example of our own code that we can add to our portfolio."

"This was a difficult project but I appreciated the work."

"Our final exam/assignment was an excellent idea. In general, especially with classes like this, doing exams in this fashion is much better for us students. Firstly, it is a more accurate representation of the student's capabilities. This is because in an exam I may have studied something but forget it and lose marks, however, in real life, if I forget something then I can just look it up and apply it. Secondly, the whole experience of doing a final assignment such as the one we just did enables us to apply what we learned in class better. Whereas, in a final, students tend to just memorize things without understanding it sometimes. Ultimately, I am very grateful that we are allowed to have a take home exam because it gave us students the opportunity to demonstrate what we have learned in class. Also, it is much less stressful for us because we have more time to prepare and do the assignment."

From ISA 330 students:

"Since the students were allowed to pick to topic their project focused on, it allowed us to have ownership and creativity on the project."

Collaboration

Students commented on how completing this assignment offered an opportunity for collaboration during the development phase, and they recognized the value in sharing completed projects with their classmates:

From CS 299 students:

Students commented on sharing their work with classmates and commenting on their work:

One student offered to his classmate, "Maybe you could connect the histogram with the data that you filter at the beginning. That way we could see the range of prices for each neighborhood."

From ISA 330 students:

ISA 330 students commented on the sharing of ideas with their peers:

"We were welcomed to branch off of the given code and discover new findings and discuss them in class."

"I enjoyed exchanging findings with classmates and trading ideas about unique ways to use Python to develop results differently."

Critical Thinking
Students commented that completing the project developed their critical thinking skills as they dealt with real-world problems creating tangible work product that demonstrated their understanding of course concepts, one that can help them as they begin their professional careers:

From CS 299 students:

"I learned that I knew more than I thought I did and was able to apply for the most part by myself without running into too many issues."

"I learned that the error I got had nothing to do with my code but was really an error with the data."

From ISA 330 students:

"The location data is incredibly messy because it is inputted by the user and it required me to really figure out how to work with the data."

Creativity
Students commented that this project allowed them to express their own creativity in choosing how to design and present their results:

From CS 299 students:

"I learned that the possibilities with code are endless and my project barely breaks the surface for what I can do."

"I learned to apply what I've learned in the class by myself and realized what I've learned … can be used in a lot of ways to portray data."

From ISA 330 students:

"I liked that I could pick the companies I wanted to study. When I started looking at the data, I realized that I was way too general and needed to narrow my focus more."

6. SUMMARY AND DISCUSSION

The PBL aspects of ownership, collaboration critical thinking, and creativity, contribute to students meeting their learning outcomes and move students from a place of engagement to a position of empowerment, motivating them to create their own original work products. While CS 299 is an introductory course without any prerequisites and ISA 330 is an advanced programming course with one prerequisite, students in both classes benefited. This implies that a PBL approach is effective for students in both beginning and more advanced courses. Students performed within reasonable expectations given their prior experience (or lack thereof) because the assignments in each course were set with reasonable expectations given the student's backgrounds. One factor influencing the effectiveness of a PBL approach is creating open-ended assignments at the appropriate level that will both challenge students and enable them to meet with success.

Students chose how they wanted to analyze and present their data, resulting in a highly personal project. When finding sample Python code from other sources, students had to understand that code so they could adapt it to their project and explain it to others. In both projects, students had to own their work even when incorporating or adapting a framework or code examples found elsewhere.

They were able to either build a solution entirely from scratch, find examples of work done by others, or review code online and adapt that code to work for their project. They shared their results and offered feedback and critique of their classmates' projects. They saw how their solutions could bring about knowledge discovery.

Students had the flexibility to pursue any avenue of their choice to analyze their data. Each student's project was different, as they had to be creative in finding the most appropriate ways using charts, graphs, word clouds, maps, or other formats to convey their findings visually. For example, in ISA 330, some students created word clouds to show hashtag frequency, while others created bar charts to present the same information.

Both projects had students interact with real datasets. They had to debug their programs in ways that required them to think critically about the context of their data. For example, some students in CS 299 reported receiving runtime errors when they chose certain combinations of data to display. While they wrote their code to filter the data correctly, they neglected to check for empty results. For example, when analyzing AirBnB listings data, some students experienced runtime errors when they tried to plot hotel rooms in the Allston neighborhood of Boston. Because the Allston neighborhood has no hotel rooms and their programs did not check for this case, their programs crashed.

PBL assignments bring additional challenges and complexities for instructors introducing them in their classrooms. The project's problem needs to be open-ended enough to provide for a variety of interpretations and solutions, but not "so open-ended" that it becomes impossible for students to grasp. The project needs to be real and manageable, without feeling contrived. Instructors must keep track of what each student is doing, and what each student is capable of doing, and determine an accurate method to assess student projects. In CS 299, students completed a short online survey after each project presentation, asking them to compare the complexity of each student's project to that of their own. This provided a baseline for assessment to the instructor, using crowd sourcing to help identify the simpler and the more complex projects. Some ISA 330 students struggled because they were trying to learn new coding skills while at the same time trying to apply mathematical analysis. They needed more examples to understand better the data analytics techniques that were applicable to their individual learning scenarios.

Students knew that they were creating a work product that they would not only present to their classmates, but also one that could become part of their professional portfolios to demonstrate their Python coding skills at interviews with potential future employers. Designing a product whose potential audience extends beyond the classroom added to their level of engagement with the project and empowered many students to explore additional ways to query and visually share their data, beyond those required for the project. Presenting their projects in class prepared students to speak confidently about their work in a future interview situation. One limitation of this project is that the authors acknowledge that a control group with a course taught using traditional methods is not part of this study. Evidence of learning is based solely on outcomes of student work and student perception of the value that they received by completing their projects.

PBL assignments offer a valuable learning opportunity for students to create software applications that demonstrate their coding skills in a meaningful way. These projects provide students the opportunity to apply their coding skills and share their work directly with others outside of the classroom. Students develop and demonstrate their skills as they work through a project, interact with real world data, evaluate their own coding abilities, and review the work of their classmates. The assignments described in this paper show two different examples of how students can have a personalized, software development experience resulting in an original data-driven Python application that they designed, developed, and implemented entirely on their own.

7. REFERENCES

Aditomo, A., Goodyear, P., Bliuc, A.-M., & Ellis, R. A. (2013). Inquiry-based learning in higher education: Principal forms, educational objectives, and disciplinary variations. *Studies in Higher Education*, *38*(9), 1239–1258. Routledge.

Astrachan, O., Hambrusch, S., Peckham, J., & Settle, A. (2009). The Present and Future of Computational Thinking. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, SIGCSE '09 (pp. 549–550). New York, NY, USA: ACM. Retrieved September 14, 2018, from http://doi.acm.org/10.1145/1508865.1509053

Baugh, J. (2011). Make it relevant and they just may learn it. *Information Systems Education Journal*, *9*(7), 14.

Bell, T., & Lodi, M. (2019). Constructing Computational Thinking Without Using

Computers. *Constructivist foundations*, *14*(3), 342–351.

Bergmann, J., & Sams, A. (2014). Flipping for Mastery. *Educational Leadership*, *71*(4), 24–29.

Bloom, B. S. (1956). Taxonomy of educational objectives. Vol. 1: Cognitive domain. *New York: McKay*, *20*, 24.

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning. *Educational Psychologist*, *26*(3–4), 369–398.

Brink, J., Capps, E., & Sutko, A. (2004). Student Exam Creation as a Learning Tool. *College Student Journal*, *38*(2).

Brown, I. W. (1991). To learn is to teach is to create the final exam. *College Teaching*, *39*(4), 150–153.

Jackson, C. (Kirabo), & Bruegmann, E. (2009). Teaching Students and Teaching Each Other: The Importance of Peer Learning for Teachers. *Working Papers*. Retrieved from https://digitalcommons.ilr.cornell.edu/workingpapers/77

Jones, Jennifer. (2016). The Student-Developed Quiz (or Exam): Scaffolding Higher-Order Thinking. *NACTA Journal*, *60*(2), 262.

McKay, Frank. (2017, October 25). 5 PBL Pitfalls to Avoid. *Edutopia*. Retrieved July 27, 2020, from https://www.edutopia.org/article/5-pbl-pitfalls-avoid

Rice, M., & Shannon, L.-J. Y. (2016). Developing project based learning, integrated courses from two different colleges at an institution of higher education: An overview of the processes, challenges, and lessons learned. *Information Systems Education Journal*, *14*(3), 55.

Rich, P. J., & Hodges, C. B. (Eds.). (2017). *Emerging Research, Practice, and Policy on Computational Thinking*. Cham: Springer International Publishing. Retrieved January 21, 2018, from http://link.springer.com/10.1007/978-3-319-52691-1

Scherz, Z., & Polak, S. (1999). An organizer for project-based learning and instruction in computer science. *Proceedings of the 4th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education*, ITiCSE '99 (pp. 88–90). Cracow, Poland: Association for Computing Machinery. Retrieved July 9, 2020, from http://doi.org/10.1145/305786.305874

Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a Phenomenology of Computational Thinking in STEM Education. *ArXiv:1801.09258 [physics]*. Retrieved January 5, 2019, from http://arxiv.org/abs/1801.09258

Sharkey, S., & Weimer, M. (2003). Learner-Centered Teaching: Five Key Changes to Practice. *Teaching Sociology*, *31*(2), 251.

Stefanou, C., Stolk, J. D., Prince, M., Chen, J. C., & Lord, S. M. (2013). Self-regulation and autonomy in problem- and project-based learning environments. *Active Learning in Higher Education*, *14*(2), 109–122. US: Sage Publications.

Treuille, A., Teixeira, T., & Kelly. (2020). Streamlit—The fastest way to create data apps. Retrieved July 22, 2020, from https://www.streamlit.io/

VanDeGrift, T. (2015). Supporting creativity and user interaction in CS 1 homework assignments. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 54–59).

Whittington, K. J. (2004). Infusing active learning into introductory programming courses. *Journal of Computing Sciences in Colleges*, *19*(5), 249–259.

Appendix 1. Assignments

Table 1. Assignments in CS 299

Assignment	Description	Concepts Introduced
Assignment #1	Calendar Calculations	Data Types and Calculations, Variables, Input, Output
Assignment #2	Laptop Configuration	Conditional Programming and Formatting Data, String Processing
Assignment #3	Mastermind Game	Loops, Lists, Strings, and Functions
Assignment #4	Numerology (calculating numerical values for words based on values of each letter)	Dictionaries, File Processing
Assignment #5	AirBnB Visualizations	Pandas DataFrames, MatPlotLib charts, StreamLit

Table 2. Assignments in ISA 330

Assignment	Description	Concepts Introduced
Assignment #1	Tools for Data Analysts	Markdown, Magic Commands, and Control Flow
Assignment #2	Representing Data	Arrays, Indexing, Slicing, Lists, and Dictionaries
Assignment #3	Programming with NumPy	Vectors, Matrix Algebra, Linear Regression
Assignment #4	Pandas Part 1	Columns/Row Manipulation, DataFrames, Loading Data, Indexing
Assignment #5	Pandas Part 2	DataFrame Operation, Sorting, Statistics, Plotting, Data Wrangling
Assignment #6	Scraping the Web	Twitter API
Assignment #7	Regression	Intro to Regression and K-Nearest Neighbor Classification

Appendix 2. Description of CS 299 Project and Examples of Student Work

CS 299 PROJECT DESCRIPTION

The second half of *CS299-1 Problem Solving with Coding* course covered these major topics:

- Lists and list comprehensions
- Dictionaries, keys, values, items, iterating
- Functions: passing parameters, returning values
- Text Files and CSV Files: reading, writing
- MatPlot Lib and various types of charts
- StreamLit.io for making interactive applications
- Pandas

Throughout the course we also talked about computational thinking ideas, and good practices for developing visualizations of data. Your final exam project is to write a Python program that shows your mastery of many of these coding concepts (and others, such as loops, strings, if statements, formatting, as needed) as you interact with data found a CSV file containing Airbnb listings from Boston. Download the Boston Airbnb listings CSV file. The data originates from http://insideairbnb.com/get-the-data.html (look for the listings.csv file for Boston).

Phase 1. Design. Due by Thursday April 30, before 12:00 pm EST
Develop two questions and two visualizations that you can create, based on this data. Examples include: What are the most expensive rentals? Is there a correlation between reviews and nightly prices? How many rentals are available in each neighborhood? You can see more sample visualizations and computations at http://insideairbnb.com/boston/. Be sure to describe how your queries will be interactive – what Streamlit user interface elements will you use? For example, you might use a slider to specify a rental price, and then a listing of homes with rental prices lower than that value. Submit your document. I will respond within 24 hours by email approving your proposed questions or making suggestions if they appear to be too complicated or too easy.

Phase 2. Build. Due by Thursday, May 5 at 7:59 AM EST (before our final exam begins)
Write a Python program to compute the answers to your questions and create the two different visualizations. Display the results using an interactive webpage coded with Streamlit.io. Place all UI controls in the left sidebar, and your visualizations in the main content area.
Your code should demonstrate mastery of these capabilities:

- At least one function that has two parameters and returns a value
- At least one function that does not return a value
- Creating and Accessing keys and values from a dictionary
- The statistics module functions (average, median, mode, etc.)
- Charts and Graphs (at least two different charts and graphs of different types, with custom legends, axis labels, tick marks, colors, other features), or map showing latitude and longitude
- User Interface and dashboard with StreamLit.io

Usual rules about writing "good" code apply:

- Make your code as modular and easy to follow as possible
- Include a docstring, comments, and meaningful variable names. If you did something "cool" in your code that you are incredibly proud of, please write a comment to point this out.
- If you referred to any online articles or other information beyond class examples, please be sure to list them as references in your code.
- Make sure the program runs and the output is correct.

Getting Help:

- This is a final exam, so please do not discuss your program with anyone other than me.
- You can ask tutors for assistance on related topics, but you cannot ask them to help debug the program you write for the final exam. For example, you can ask tutors to help review examples of how to create bar charts in Python (in general), but you cannot ask them to help you debug a bar chart you might create for this exam using the Airbnb data.

Phase 3. Present and Review.

In Class on Tuesday, May 5 at from 8:00 am to 10:00 am (During Scheduled Final Exam)

You will present your project for approximately 5 minutes during the final exam period, and your classmates will provide feedback to you in an online form. For students in other time zones, or who cannot attend the class session live, please create a short (fewer than 5 minutes) video in which you describe your code and show us how it runs. Upload the video (unlisted to YouTube) or to your cloud storage, and send me a link.

Grading:

- Design Proposal and Document – 10 %
- Code – 80%
- Presentation – 10 %

EXAMPLES OF STUDENT WORK

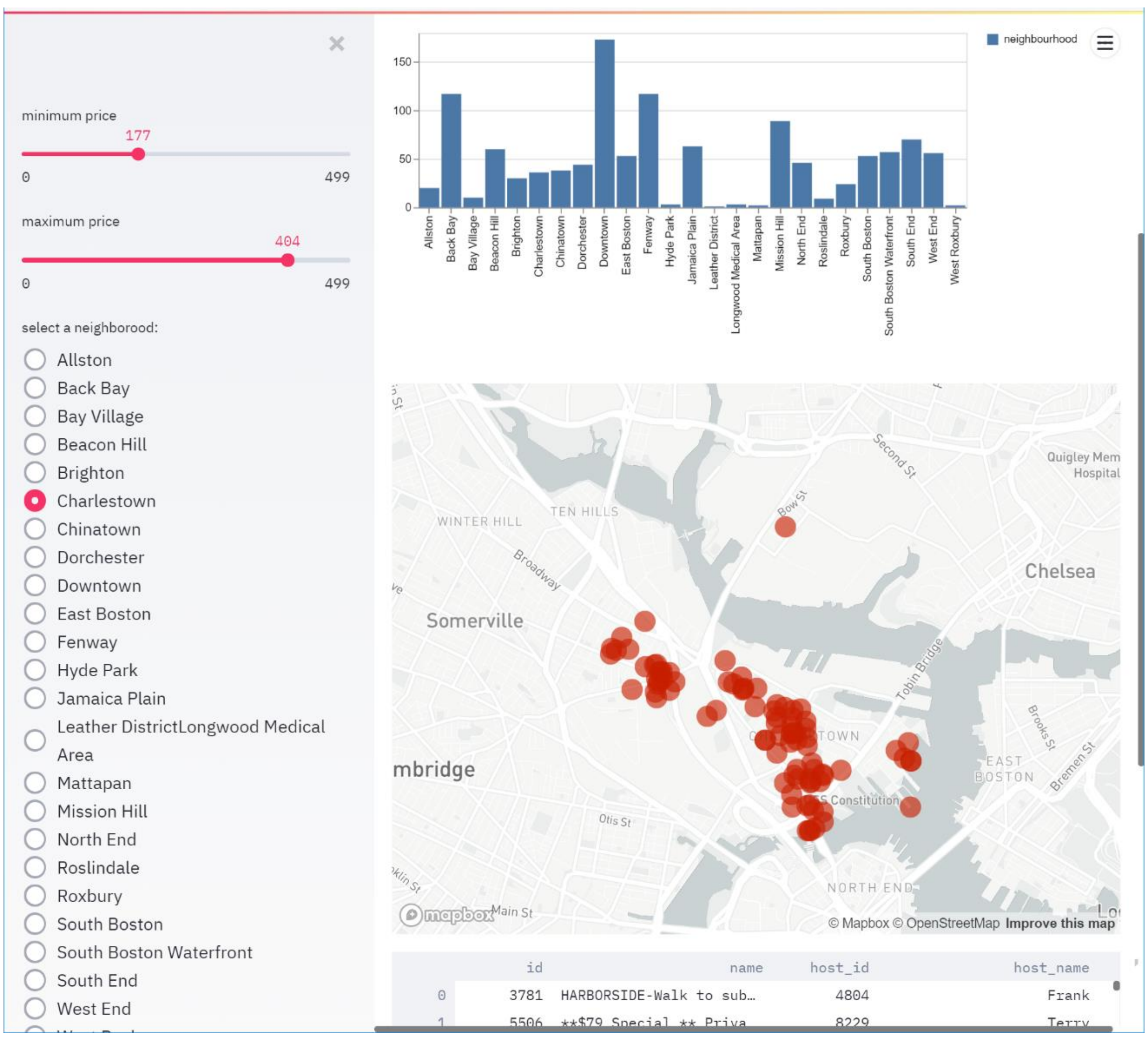

Figure 1. Chart and Map

In this visualization of Boston AirBnB listings data, a student chose to create a page displaying properties within a specified price range in a chosen neighborhood. Streamlit controls make the query interactive by allowing the user to select these values and then the chart and map update to reflect the new results from the query.

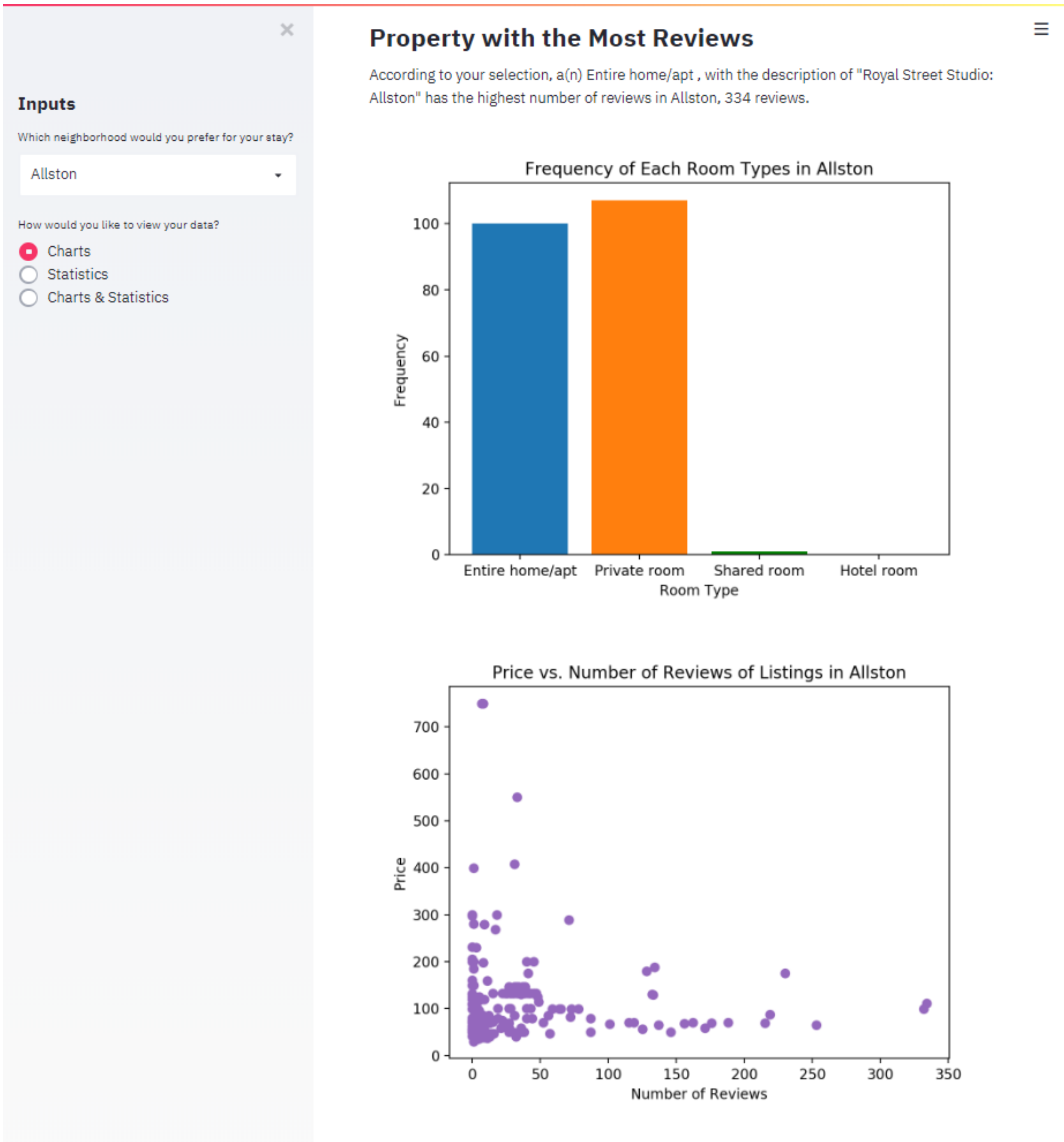

Figure 2. Bar chart and Scatter Plot

In this visualization of Boston AirBnB listings data, a student chose to create a page displaying information about properties with the largest number of reviews. Streamlit controls make the query interactive by allowing the user to select the neighborhood from a dropdown list and the desired output, and then the charts update to reflect the new results from the query.

Appendix 3. Description of IS 330 Project and Examples of Student Work

ISA 330 PROJECT DESCRIPTION

In this individual project, you will learn how to collect and compare social media data on two companies/products/topics of your choosing.

Phase 1: Set Up Account (In Class). You will each set up a developer accounts for Twitter (developer.twitter.com). Having a developer account allows you to gather a vast amount of Twitter data. Each of you will have your own account that you can use in future classes or even with your future employer. You will learn the different between the various APIs available to you and the difference between "pulling" historical data and "listening" for real-time data. By the end of this phase, you will have an active Twitter account gathering data.

Phase 2: Topic Identification (Both In-Class and Outside of Class). Once you have your Twitter account set up you will decide on a topic to examine. These topics could be two different companies (Microsoft versus Apple, Adidas versus Nike, etc.), two different products (Doritos versus Fritos, Corvette versus Mustang, etc.), or even two different topics (Black Lives Matter versus Blue Lives Matter, Pro-Choice versus Pro-Life, etc.). You will start (in class) by using the Twitter front end to explore your topic and make sure your hashtags are appropriate for data gathering. You will want a set of hashtags that are both appropriate for your topic (and without non-applicable data) as well as popular enough so you can gather several thousand Tweets over a several week period (i.e. #Trump and #Clinton would probably give you too much data while #ILoveGreenMarbles and #IHateGreenMarbles will not give you enough data). Make sure you confirm with the professor your topic before you start to acquire the data.

Phase 3. Acquire Data (Outside of class). Once you have your appropriate hashtags identified then you will run the Twitter streamer on your own laptops. It is important that you keep your laptop on and running during your period of study.

Phase 4: Analyze Data. We will start with a basic sample Jupyter Notebook and a dataset collected by the professor. You will segment the data into two different DataFrames and perform some basic text analysis on the Tweets. We will discuss how to find other sample code. You will explore different avenues of data exploration and share that knowledge with the class through mini informal presentations. You may share ideas and code with each other, but you are responsible to fully understand and customize any code you obtain from other sources (as well as cite that source in the notebook).

Phase 5: Reflect and Summarize. The final deliverable includes:

1. Executive summary. A 2-3 page summary of your findings. Please present this in such a way to be consumable by management (or other non-technical people).
2. Jupyter Notebook. Submit your fully documented Jupyter Notebook. It should contain enough comments throughout to walk another developer through your process. Make sure you also include lessons learned and any analysis that you did even if you did not include it in your executive analysis. As an analytics professional, you often may do exploratory work that has disappointing results, and it is important to document this so others learn from these trials.
3. Data. Submit your .JSON data file that you collected during the study. You can use this data file to rerun your entire Jupyter Notebook.
4. Presentation. Give a brief presentation to your fellow classmates. Imagine you are in a job interview and your interviewer asks you to discuss the project – this is your presentation. In about 5 minutes, walk through the notebook, and discuss your project.

Grading:

- Informal class presentations and participation: 20 %
- Fully Documented Code – 50%
- Executive Summary - 20%
- Presentation – 10 %

EXAMPLES OF STUDENT WORK

Using Jupyter Notebooks allowed students to create a work product that not only demonstrated their coding skills but also presented the findings in a single notebook. In this example, the student uses various techniques to explore the differences in text content of Twitter messages mentioning "Spotify" versus "Pandora", two popular music-streaming services.

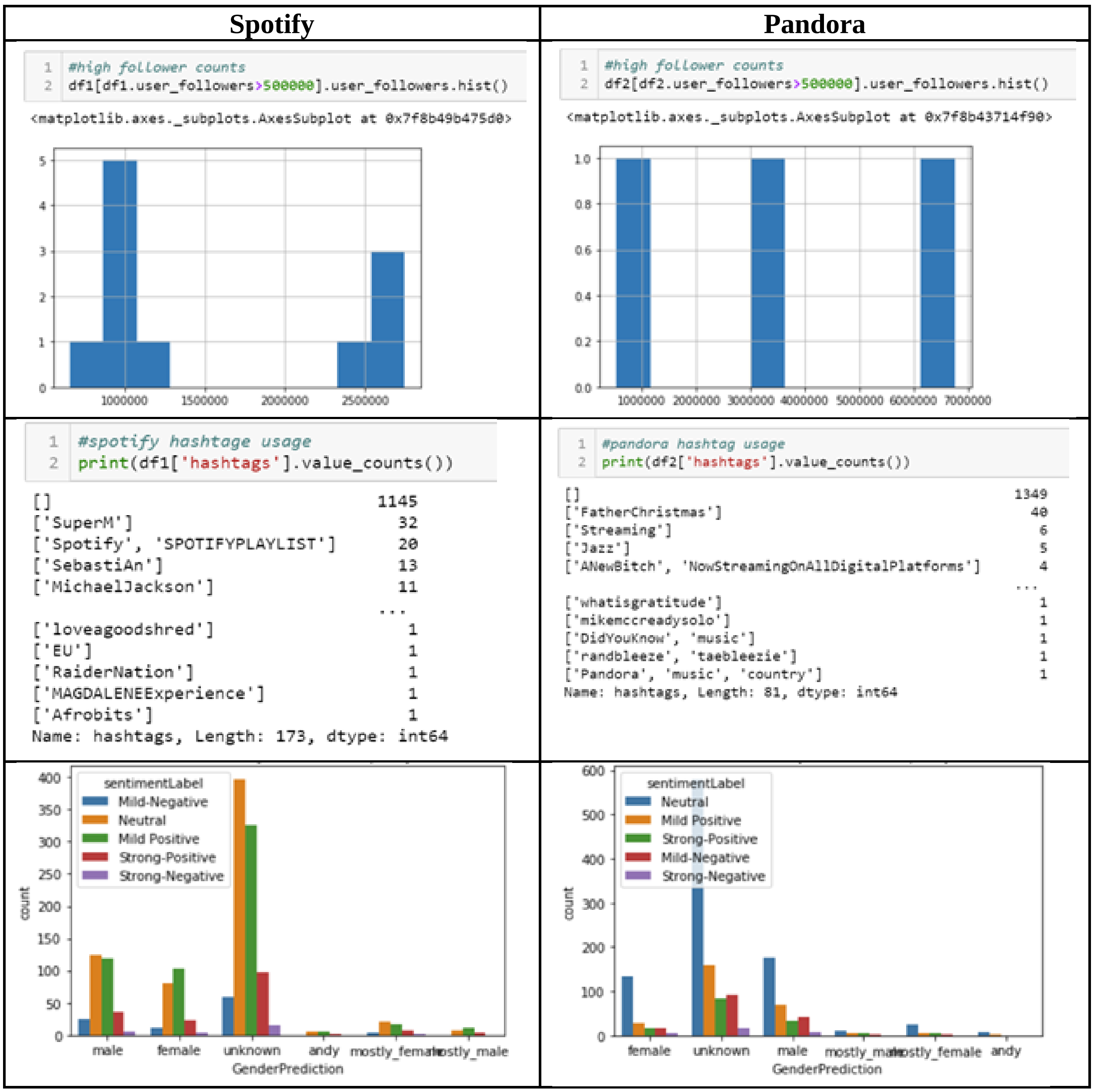

Figure 1. Charts comparing Tweets mentioning Spotify versus Pandora

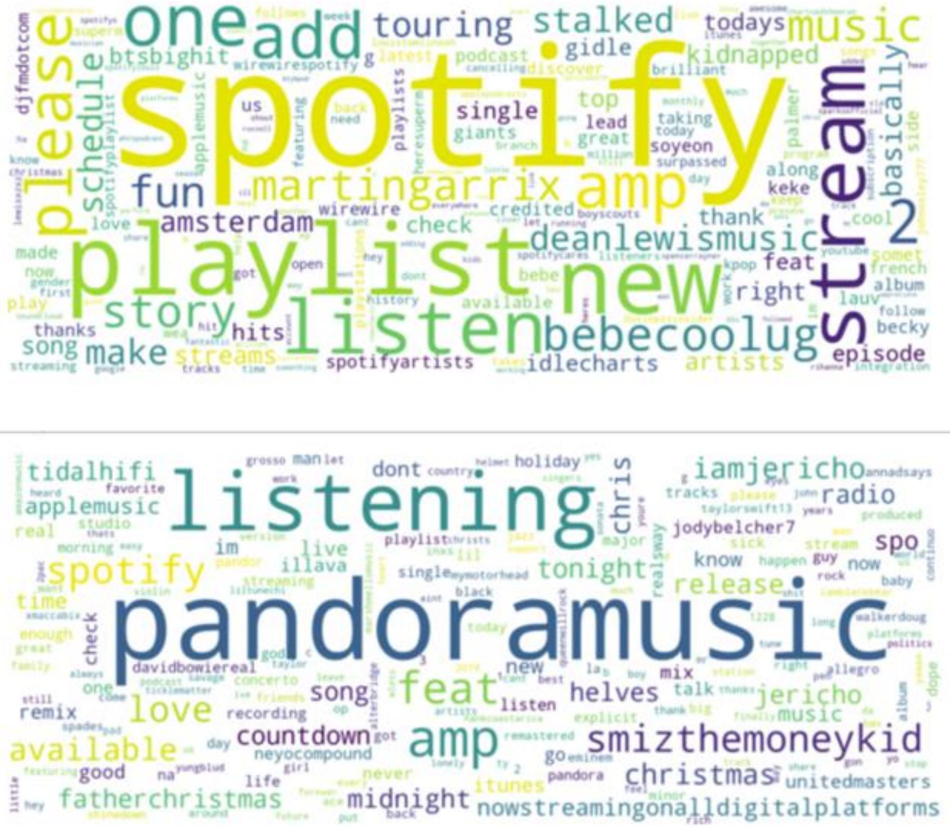

Figure 2. Word Clouds showing text data of Spotify versus Pandora.

Students enjoyed the quick visualization obtained using word clouds. Here the student was able to hone in quickly that playlists related to Spotify had more discussion on Twitter than playlists talking about Pandora, perhaps because Pandora playlists are a premium service. Word Clouds are popular among students and provide an easy technique for summarizing unstructured text data.