INFORMATION SYSTEMS EDUCATION JOURNAL

Volume 18, No. 4
August 2020
ISSN: 1545-679X

In this issue:

4. Using Goal Setting Assignments to Promote a Growth Mindset in IT Students
David M. Wood, Miami University Regionals

12. Liberating Legacy System Data with Rails, Intelligent Use of Conflict Data with Automated Class Scheduling Tools
Stuart L. Wolthuis, Brigham Young University – Hawaii
Christopher Slade, Brigham Young University - Hawaii

22. Undergraduate Business Analytics and the overlap with Information Systems Programs
Wendy Ceccucci, Quinnipiac University
Kiku Jones, Quinnipiac University
Katarzyna Toskin, Quinnipiac University
Lori Leonard, The University of Tulsa

33. Lizards in the Street! Introducing Cybersecurity Awareness in a Digital Literacy Context
Mark Frydenberg, Bentley University
Birgy Lorenz, Tallinn University of Technology

46. Academic Entitlement Beliefs of Information Systems Students: A Comparison with Other Business Majors and An Exploration of Key Demographic Variables and Outcomes
Scott J. Seipel, Middle Tennessee State University
Nita G. Brooks, Middle Tennessee State university

59. An Assignment a Day Scaffolded Learning Approach for Teaching Introductory Computer Programming
Deepak Dawar, Miami University
Marianne Murphy, Miami University

74. Learning How to Teach: The Case for Faculty Learning Communities
David Gomillion, Texas A&M University
Aaron Becker, Texas A&M University
Jordana George, Texas A&M University
Michael Scialdone, Texas A&M 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 (http://isedj.org). Our sister publication, the Proceedings of EDSIGCON (http://www.edsigcon.org) 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 30%), 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 EDSIG who perform the editorial and review processes for ISEDJ.

2020 Education Special Interest Group (EDSIG) Board of Directors

INFORMATION SYSTEMS
EDUCATION JOURNAL

Using Goal Setting Assignments to Promote a Growth Mindset in IT Students

David M. Woods
Woodsdm2@miamioh.edu
Computer & Information Technology Department
Miami University Regionals
Hamilton, OH 45011

Abstract

This paper explores how goal-setting activities in a course were used to promote a growth mindset in students. Research shows many benefits for students with a growth mindset that emphasizes learning and addressing challenges by focusing on effort and process rather than judgments about success or ability. Activities designed to prompt students to improve general skills that would make them better students and prepare them to be lifelong learners were introduced in two upper level IT courses. The activities were designed to promote a growth mindset by focusing on the efforts made and processes used rather than the outcomes. Assessment of the activities found that students demonstrated a growth mindset in their work, saw clear value in the activities, and made progress in improving specific skills.

Keywords: Growth mindset; goal setting; life-long learning; IT education; pedagogy

1. INTRODUCTION

We teach, but our real goal is for students to learn and prepare for success in life. For instructors, there are many parts to this. We must learn about the latest technologies and update classes to add technologies that employers seek. We need to adjust content and delivery to address the move to hybrid and online courses. Ever present concerns about retention and completion statistics mean that we must ensure that students are actively engaged in our programs and institutions. In addition, all fields, but especially IS/IT must prepare students to be lifelong learners.

At the center of all of these efforts are our students, with the instructor in the classroom as the main person engaged in helping the student learn. The instructor can't do it all, so institutions provide a range of services to support students – learning assistance centers, tutoring, study skills courses, etc. Some students take advantage of these support services, and others may not need them, but there are students who need to improve their skills but fail to make use of these services.

This failure to develop needed skills is puzzling. Students are given clear feedback about skills they need to improve – writing, time management, etc. – along with information about where they could find assistance, but no improvement is seen. Discussions with students offered many explanations for not developing these skills that would help them in all of their courses. Two themes stood out. First were the students who knew that poor skills were limiting their ability to succeed, but did not feel that they could improve these skills, demonstrating the fixed mindset identified by Dweck (2016). Another theme was students who set goals for improvement but struggled to take action and make progress towards their goals.

This information prompted thought about expanding course activities and assignments to help students foster a growth mindset, set goals, and make progress towards achieving these goals. These efforts serve many purposes, but fundamentally, the goal is to help students improve their skills as students. Improved skills will help them learn more in a specific course, help them in later courses, and make them more confident in their ability to complete their degree program. After they graduate, employers will benefit from new employees who can take ownership of planning and executing the learning and improvement necessary to remain valuable employees.

2. MOTIVATION

Mindset

Work looking at individuals' attitudes has identified two mindsets that affect how people respond to the challenges they encounter in their life (Dweck, 2016). People with a fixed mindset believe that their "qualities are carved in stone" (Dweck, 2016, p6) and feel they have a fixed intelligence and personality. People with a growth mindset believe that these qualities can be developed through their efforts, strategies, and the help of others.

The two mindsets drive significant differences in an individual's behavior and how they react to challenges. People in the fixed mindset feel they are constantly being evaluated – are they smart or dumb, will they succeed or fail, will they win or lose? A challenge is seen as a test where they must succeed or fail. They focus on the judgment and may ignore feedback about how to improve their performance. If they do not succeed in their first effort, they may give up. With the growth mindset, people are not interested in proving themselves, but rather improving themselves. The person with the growth mindset feels smart when "learning something over time: confronting a challenge and making progress (Dweck, 2016, p 24). The person with a growth mindset seeks to overcome a challenge by working harder, trying different strategies, and seeking help from others.

There are connections between mindset and the concept of grit, defined as "perseverance and passion for long-term goals" (Duckworth and Peterson, 2007, p 1087)" which has been shown to predict success factors beyond those predicted by IQ. Duckworth (2013) identified the growth mindset as "the best idea I've heard about building grit in kids."

A growth mindset seems ideal for learning, and studies have explored the impact of mindset in education. A recent study by the Center for Community College Student Engagement explored many aspects of mindset. One finding was that "More students have fixed mindsets for math then for either English or overall intelligence" (CCCSE, 2019, p6). The growth mindset also correlates with higher GPAs in both math and English. These findings could affect student success and retention and have specific interest for IS/IT educators since math is seen as a closely related field. Another finding from the study is a relationship between maturity and mindset, with non-traditional age students showing more optimism when facing challenges.

Research on connections between mindset and poverty shows how a growth mindset helps poor students overcome some of the obstacles they face. Research on a national scale looked at the mindset of public school students in Chile (Claro, Paunesku, Dweck, 2016). This work found that mindset and socioeconomic factors are both strong predictors of academic achievement. The study found that a growth mindset was more common with students from higher income families. The finding that "students in the lowest 10th percentile of family income who exhibited a growth mindset showed academic performance as high as that of fixed mindset students from the 80th income percentile" highlights the potential value of promoting the growth mindset (Claro, Paunesku, Dweck, 2016, p8664).

The mindset of faculty can have a significant impact on students. Recent work that looked at a sample of 150 STEM instructors and 15,000 students found that students in courses taught by instructors with a fixed mindset earned lower grades (Canning, Muenks, Green, & Murphy, 2019). In addition, while students from underrepresented minorities had lower average grades than white or Asian peers, the study found that this racial achievement gap was twice as large in courses taught by instructors with a fixed mindset. This work also reviewed course evaluations and found that students were less motivated in courses taught by faculty with a fixed mindset, and were less likely to recommend a course taught by an instructor with a fixed mindset.

Dweck notes that "in truth we're all a mixture of the two" mindsets, and that various events or situations may trigger a specific mindset (2016, p 211). Grant and Dweck (2003) performed five studies that looked at the impact of goals on mindset. Ability or performance goals predict fixed mindset results where student performance and engagement suffer in the face of a challenge. In contrast, goals focused on learning and gaining new knowledge predict growth mindset behavior - "active coping, sustained motivation, and higher achievement in the face of a challenge (Grant and Dweck, 2003, p541)." This finding shows the importance of focusing on learning goals rather than performance, providing feedback focused on effort, and offering processes to support students' efforts.

Several efforts have explored applying mindset thinking to technology courses (Murphy and Thomas, 2008; Cutts, Cutts, Draper, O'Donnell, and Saffrey, 2010; Lovell, 2014; Payne, Babb, and Abdullat 2018). An obvious starting point is an initial programming course, which can present students with many unexpected challenges along with the potential for technology-generated feedback, including syntax errors, compiler errors, and run-time errors, that are presented in a fixed mindset type success/failure format. One study found that teaching students about mindset and providing growth mindset motivated feedback to students during a six-week period had a positive impact on student's mindset and test scores (Cutts et al. 2010).

Goals Setting

With the value of processes like goals in supporting a growth mindset, it is interesting to look at research on goal setting. Research finds that goal setting in the workplace has a positive impact on employee engagement, workplace optimism, and individual performance – signs of a growth mindset (Medlin and Green, 2009).

Research on the use of goal setting in the classroom also shows benefits. When students in a management course used a goal setting worksheet to develop goals for a group project, instructors found that students actively used the goals to improve project quality and team performance (Lawlor and Hornyak, 2012).

Both of these studies found value in formal, structured goal setting processes. Lawlor & Hornyak specifically used the SMART goal approach. The first published discussion of the SMART goals defined the acronym as Specific, Measurable, Assignable, Realistic, and Time-related (Doran, 1981). Since then, several useful variations have developed (SMART criteria, n.d.), including a format that uses Achievable in place of Assignable (SMART Goals, n.d.).

3. GOAL SETTING ACTIVITIES

How can we promote a growth mindset in students and encourage them to develop skills that make them better students, and in the future, better employees? The growth mindset's focus on effort and process suggests exploring the use of goal setting as a specific process to support the growth mindset. Goal setting activities designed to promote a growth mindset and help students build general skills, rather than skills specific to one course, were developed. The goal setting activities were used in two different upper-level IT courses but were not tied to specific course projects. In addition to the goal of promoting a growth mindset, a second goal was to measure student perceptions of the goal setting activities to guide further use and development of the activities. Students in both courses are a mix of traditional age and older, non-traditional students, with many students working part-time or full-time while taking courses.

Personal Improvement Project

The Current Practices in Information Technology course is the first course in a three-semester self-directed capstone experience. For their capstone, students use technology to develop and implement a solution to a specific problem. During the first capstone course, students work individually to research potential capstone project ideas. In addition to learning about a problem, the research often involves exploring technologies and tools for potential solutions.

During the semester, students complete four three-week long research projects. Each project includes assignments for a project proposal, in-class project pitch, intermediate work product, final work product, reflection, and in-class project presentation. A challenge of this course is that students must take ownership of planning and managing their projects. Additionally, oral and written communication skills are important for the project pitch, in-class presentation, and final project report. A Personal Improvement Project activity was developed to provide a process to promote a growth mindset in the development of the soft skills used in this class.

The activity had three graded assignments during the course of the semester. In the second week of the semester, students submitted a proposal setting a goal to improve a specific non-technical skill along with a discussion of why they chose the specific skill. The proposal assignment prompted students to think about how they would measure and report on their progress in later assignments. The assignment also provided examples of soft skills and potentially useful on-campus and online resources.

In the middle of the semester, students completed a status check assignment. Using a growth mindset approach, the assignment prompted students to think about effort and process. Students submitted a reflection about what they had done, whether they wanted to make updates to their initial goal, whether they needed help to work towards their goal, and their plans for working on their goal during the rest of the semester.

The final assignment was an end of the semester wrap up. Again, students reflected on their work to achieve their goal and discussed whether they would continue working on the goal or add a new goal.

Student Performance Planning

Goal setting was also used in a course that covers IT strategy and management. This course covers a wide range of topics, but one specific learning outcome covers the management of IT staff. Material supporting this learning outcome includes hiring, promotion, and employee performance planning and evaluation. To help students understand employee performance planning and appraisal, student performance planning and evaluation activities were developed. These activities are spread throughout the semester and provide processes to promote a growth mindset.

The course text uses a novel like format to follow a business leader unexpectedly thrust into the role of Chief Information Officer (CIO) at a fictional company (Austin, Nolan, and O'Donnell, 2016). The book starts with this leader moving into his new role, similar to students starting a new class. In the first week of class, a discussion of goal setting and performance planning for the main character in the book is used to support a discussion of goal setting and performance planning for students. The discussion introduces the SMART goal concept, along with examples of writing SMART goals.

In the first performance planning assignment, students develop a student performance plan for their work in the course during the semester. Students are provided a performance planning template and develop goals organized into three groups, with examples provided for each group:

- General Student Activities – activities a student might do in any course they take.
- Achievement of course learning outcomes – activities to help the student achieve this specific course's learning outcomes.
- Teamwork – goals to support team assignments in the course.

The class has several team assignments, and students work in the same teams for all assignments. While students are developing their teamwork goals, the teams are also working on a team organization and planning assignment and are encouraged to connect their personal teamwork goals with the plans developed by their team.

In the middle of the semester, students complete a two-part midterm performance assessment. First, students assess their progress for at least two goals in each of the three groups in the performance plan. Students are encouraged to submit data to support their self-assessment. For example, one student with a goal about the number and quality of classroom contribution submitted a spreadsheet documenting and assessing each of their classroom contributions. Secondly, students reflect on how creating and following a performance plan has helped their overall performance in the class, with specific discussion of:

- Their execution of activities to support their goals.
- Whether they wrote the right goals
- How they might improve their goals or their work to achieve them.

The final assignment in the performance planning activities was an end of semester assessment of the performance plan. The assignment had two parts. The first was the same as the midterm assessment – an assessment of progress on the goals, ideally supported with data. The second part asked students to reflect on how the performance planning activities had helped their performance in the class. Students also discussed how they might use performance planning, including specific goals, in future courses or a work environment. All of the performance planning assignments contributed to the student's final course grade.

4. ASSESSMENT

Two methods were used to assess the two goal setting activities. The student submissions were reviewed to determine how students engaged in the activities and assess their mindset. An end-of-semester survey collected data about student views on the value of the activities and effort required.

Personal Improvement Project

Students in two successive semesters completed the personal improvement project. The course enrollment was thirty-one (31) in the first semester and eleven (11) in the second. In both semesters, student goals covered a range of topics. The most popular covered time management (procrastination, scheduling, and work/life balance), self-care (meditation, exercise, stress management), and communication (writing and public speaking).

The end of semester wrap up assignment was reviewed to assess what mindset students exhibited in discussing their work on the personal improvement project. Of the thirty-three (33) students who consented to participate in the research, all but one completed this assignment.

The student submissions provided clear discussions of what the student learned and the impact of the projects. Students showed pride and even surprise in what they were able to accomplish. All of the student discussions addressed one or more concepts associated with the growth mindset. These included the effort they made, the processes they used, and the progress they made. Many also discussed plans to continue work on their goal.

Nine of the responses included judgments or similar content associated with a fixed mindset. Four of these were positive – noting the accomplishment of a goal or the success of the project. The other five submissions used terms reflecting disappointment, failure, or scoring their effort poorly. At the same time, all of these submissions also included discussion of the effort and progress that the student had made in working on their goal, and all of these students exhibited signs of a growth mindset by discussing how they would continue to work on their goal.

Students completed a short, anonymous survey on the last day of class. In the first semester, twenty-three (23) of the students completed the survey (74 % response rate), and in the second semester, ten (10) students completed the survey (91 % response rate).

The first four questions used a 5 point Likert scale, asking students to agree or disagree with the statements:

1. I felt that the personal improvement project helped me improve my skills as a student.
2. I felt that the mid-semester status check on the personal improvement project helped me assess how I was doing with my improvement project.
3. I saw the value of the personal improvement project for improving my work as a student.
4. The feedback provided by the instructor encouraged my efforts to work on this project.

For all four questions, the average response was at least 4.0, with at least 75% of students agreeing or strongly agreeing, indicating that students saw clear value in the activities and that instructor feedback promoted a growth mindset.

The remaining questions used a 7 point Likert scale for students to rate (not much to very much):

5. How effortful was it for you to work on your personal improvement project?
6. How much did the personal improvement project help your ability to complete the research projects in the course?
7. How much did you enjoy the personal improvement project?
8. How much would you like to do a similar personal improvement project in future courses?

For question 5, the average was 5.2, showing the project required some effort. The results from the remaining questions were all positive (4.8 – 4.9) showing that the work was beneficial and enjoyable. Question 8 had the widest distribution of answers, with ten (30 %) very much wanting to do a similar project in a future course, but also with thirteen (39 %) unsure.

Student Performance Planning

The student performance planning activities were used in a recent semester of the course with twenty-four (24) students. Goal setting and the assignment to write a student performance plan were introduced in the first week. The SMART goal concept was also introduced with in-class discussions and examples and supporting online material. Students identified good goals but struggled to document them as SMART goals. The main issues were goals that were not measurable or were not specific. To address these problems, specific feedback was provided. Students were encouraged to discuss their goals with the instructor and allowed to resubmit their student performance plan.

It was a pleasure to read the student reflections submitted with the final performance evaluation. The reflections showed that students had made clear progress in accomplishing their goals. The student reflections also showed that students had put significant effort into working on their goals. For many students, their discussion of how goal setting had helped them improve as students matched what was observed in their class participation and submitted assignments.

All of the student discussions addressed the effort and process concepts linked to the growth mindset. All of the students mentioned goal setting and performance planning as a valuable process. Many also mentioned how the performance plan motivated them to be accountable to make an effort to work on the goals. Two students did make clear judgments that they were not able to achieve their goals, but their discussions focused on lack of effort, a sign of a growth mindset, rather than lack of ability, a sign of a fixed mindset.

An anonymous end of semester survey was used to collect information about several class activities, including the student performance plan, and sixteen students completed the survey (67 % response rate).

The first five questions on the survey used a 5 point Likert scale, asking students to agree or disagree with the statements:

1. I saw the value of the performance planning activities for learning how to write good performance goals for the course.
2. I saw the value of the performance planning activities for evaluating my own performance.
3. I saw the value of the performance planning activities for planning to improve my own performance.
4. I saw the value of the performance planning activities to prepare me for performance planning I might do in a professional workplace.
5. I felt that the performance planning activities engaged me in thinking about how to improve my performance as a student.

For questions 1 – 4, a clear majority (69 – 88%) of students agreed that they saw value in the different goals of the activities. For question 5, the majority (69%) also agreed that the activities prompted them to engage in the process of self-improvement.

The remaining questions about the performance planning activities used a 7 point Likert scale for the students to rate (not much to very much):

6. How effortful was it for you to write your initial performance plan?
7. How effortful was it for you to complete your mid-term performance evaluation?
8. How effortful was it for you to complete your final performance evaluation?
9. How much did you enjoy the performance planning activities?
10. How much did you learn about setting good goals?
11. How much did you learn about a planning process for improving your work in a class or similar long term activity?
12. How much would you like to do similar performance planning activities in future courses?

For questions 6 – 8 about the effort for the activities, averages were 4.1 – 4.5, with writing the initial performance plan requiring the most effort. The response for question 9 about the enjoyment was overall neutral – 4.1. For questions 10 and 11, the averages show students learned about both processes that could be used to support a growth mindset - goal setting (5.1) and performance planning (4.9). For the final question, 50% of the students wanted to do similar activities in future courses. Several students in the course were about to graduate, which may have affected the responses to this question.

5. CONCLUSIONS AND FUTURE PLANS

Students in both classes demonstrated a growth mindset and saw value in the goal setting activities. The presence of a growth mindset is not clearly linked to the class activities, but both activities met the goals set when the activities were developed. The activities were beneficial to the majority of the students, did not require too much effort, and were well received by students. From the instructor’s perspective, the time required to develop and grade the assignments was minimal, and the student submissions provided good insight into the effort and progress students were making. There was also clear evidence that students appreciated and acted on the feedback they received.

The results are encouraging, supporting further efforts to use goal setting in the future. The design of these activities to focus on general, non-technical skills should allow use in a wide range of courses.

Goal setting activities are being developed for two introductory courses. These will use a version of the personal improvement project with more frequent status updates. One of the introductory courses also includes first-year experience content that all new students at the university are required to take, which will provide an excellent opportunity to discuss the mindset and SMART goal concepts.

Goal setting is just one process that can support a growth mindset. Further work will review other aspects of the course environment to identify additional opportunities to encourage the growth mindset.

6. REFERENCES

Austin, R. D, Nolan, R. L., & O'Donnell, S. (2016). The Adventures of An IT Leader. Harvard Business Review Press, Boston.

Canning, E. A., Muenks, K., Green, D. J., & Murphy, M. C. (2019). STEM Faculty Who Believe Ability is Fixed Have Larger Racial Achievement Gaps and Inspire Less Student Motivation in Their Classes. Science Advances, 5(2), EAAU4734.

Center for Community College Student Engagement (CCCSE) (2019). A mind at work: Maximizing the relationship between mindset and student success. The University of Texas at Austin, College of Education, Department of Educational Leadership and Policy, Program in Higher Education Leadership, Austin, TX. Retrieved from http://www.ccsse.org/center/NR2019/Mindset.pdf

Claro, S., Paunesku, D., & Dweck, C. S. (2016). Growth Mindset Tempers the Effects of Poverty on Academic Achievement. *Proceedings of the National Academy of Sciences*, 113(31), 8664-8668.

Cutts, Q., Cutts, E. Draper, S., O'Donnell, P., & Saffrey, P. (2010). Manipulating Mindset to Positively Influence Introductory Programming Performance. In *SIGCSE 10: Proceedings of the 41st ACM technical symposium on computer science education, Milwaukee (WI), USA, 10-13.03.2010*. Association for Computing Machinery, New York.

Doran, G. T. (1981). There's A S.M.A.R.T. Way to Write Management's Goals and Objectives. *Management Review*, 70(11), 35.

Duckworth, A. & Peterson, C. (2007). Grit: Perseverance and Passion for Long-Term Goals. *Journal of Personality and Social Psychology*, 92(6), 1087-1101.

Duckworth, A. (2013). Grit: The Power of Passion and Perseverance. *TED Talks Education*, retrieved from https://www.ted.com/talks/angela_lee_duckworth_grit_the_power_of_passion_and_perseverance

Dweck, C. (2016). *Mindset: The New Psychology of Success, Updated Edition*. Ballantine Books, New York.

Grant, H. & Dweck, C. S. (2003). Clarifying Achievement Goals and Their Impact. *Journal of Personality and Social Psychology*, 85(3), 541-553.

Lawlor, K. B. & Hornyak, M. J. (2012). SMART Goals: How the Application of SMART Goals can Contribute to Achievement of Student Learning Outcomes. Developments in Business Simulation and Experiential Learning, 39, 259-267.

Lovell, E. (2014). Promoting Constructive Mindsets for Overcoming Failure in Computer Science Education. In *ICER '14 Proceedings of the tenth annual conference on International computing education research.* Association for Computing Machinery, New York.

Medlin, B. & Green Jr., K. W. (2009). Enhancing Performance through Goal Setting, Engagement, and Optimism. *Industrial Management & Data Systems*, 109(7), 943-956.

Murphy, L. & Thomas, L. (2008). Dangers of a Fixed Mindset: Implications of Self-theories for Computer Science Education. In *ITiCSE 08: Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education: Madrid, Spain, June 30-July 2, 2008*. Association for Computing Machinery, New York.

Payne, K. C., Babb, J., & Abdullat, A. (2018). Reflections on Applying the Growth Mindset Approach to Computer Programming Courses. In *2018 Proceedings of the EDSIG Conference Norfolk, VA USA*, 4(4641).

SMART criteria (n.d.) In *Wikipedia*. Retrieved July 9, 2019, from https://en.wikipedia.org/wiki/SMART_criteria

SMART Goals: How to Make Your Goals Achievable. (n.d.). Retrieved from https://www.mindtools.com/pages/article/smart-goals.htm

Liberating Legacy System Data with Rails, Intelligent Use of Conflict Data with Automated Class Scheduling Tools

Stuart L. Wolthuis
stuart.wolthuis@byuh.edu

Christopher Slade
christopher.slade@byuh.edu

Faculty of Mathematics and Computing
Brigham Young University-Hawaii
Laie, HI 96762

Abstract

In this research project we describe the legacy software landscape, its current state, and challenges associated with aging information systems and access to its data. We briefly describe the popularity of dynamic languages and how a specific dynamic programming language, Ruby on Rails (RoR or Rails), is used to create a system to extract data from a legacy system to increase efficiency and productivity in an academic class scheduling system. As an example, we describe, first, how a system developed in Rails, called Class Scheduler, pulls data from a legacy student management system (MAPPER) developed in Tcl (pronounced "tickle") and uses this data to vastly increase the efficiency of the scheduling process and, second, how it reduces conflicts in class schedules. We discuss the advantages of automatically extracting and processing the data from the legacy system and the limitations associated with this process.

Keywords: Ruby on Rails; RoR; Rails; legacy systems; legacy data; software engineering; programming; scheduler; class scheduling;

I. INTRODUCTION

With the advent of the cloud and the use of Software as a Service (SaaS) an enormous amount of data produced and stored in legacy information systems can be left behind and left inaccessible unless solutions to extract this data are realized. Without modernization, users can keep using legacy systems as they exist and hope the hardware and operating systems providing access to their valuable data continue to function. As a benchmark to the state of large information systems a 2016 Government Accountability Office (GAO) report to Congress stated that in 2015 $61.2 billion was spent on operations and maintenance of current (legacy) systems while $19.2 billion was spent on development, modernization, and enhancement (Powner 2016). Another insightful indicator reported by the GAO is that the amount of IT spending on development, modernization, and enhancement from 2010 to 2017 declined by $7.3 billion, a 28% reduction. This implies that enhanced digitization, which may equate to access of legacy data, is not a current priority for these maturing systems due, in part, to its very costly nature given the three imperatives of data migration: don't interrupt current business processes, maintain data consistency, and effort and cost should be minimized (Martens, Book, Gruhn, 2018).

Aging government information systems include two master files (individual tax files and business tax files) in the Department of the Treasury that are approximately 56 years old with no specific plans for updates (Powner, 2016). Also cited in this report is the fact that the nuclear command and control system is 53 years old and still runs on an IBM Series/1 Computer with an 8-inch floppy drive.

Systems that manage inmates in prisons, including their security, custody levels, and work assignments is over 35 years old and your SSN is managed by a system that is 31 years old (Powner, 2016).

Additionally, the 12 government agencies in this report indicated using unsupported operating systems; 5 reported using 1980s and 1990s Microsoft operating systems, with no support from the vendor for over a decade.

Any organization that continues to use antiquated technology systems must pay a premium for staff or contractors with the right knowledge to support and maintain legacy systems (Powner, 2016). For example, the author of this paper personally managed the software development of a product improvement program in 1993 with a defense contractor that required pulling an employee out of retirement to change 160 lines of code in a complex system originally written in Fortran 66.

A final note on government legacy systems: the Department of Commerce runs a system providing warnings to the public and emergency managers with several obsolete operating systems: Windows Server 2003 no longer supported by the vendor and a version of Oracle no longer fully supported by the vendor (Powner, 2016). These systems observe meteorological incidents that could generate a tsunami or hurricane.

Both the data contained in legacy systems and the systems themselves require expertise and innovation to maintain their integrity. In this paper we will specifically address the challenge of accessing and using legacy data.

Background

Efforts to quantify the amount of legacy data in the world, or more specifically, to quantify the amount of valuable, relevant, or useful data in our world, appears to be the subject of a few blogs and white papers, but is woefully neglected in the annals of scholarly research. This may simply be due to the modernity of this situation. Our review of literature found very little direct evaluation of the legacy data problem in academia. The GAO is relegated to review information systems providing public services, those maintained and paid for by federal or state budgets, and seems to be the only entity addressing the white elephant in the room.

An estimate of the amount of data created reports a "truly mind-boggling" 2.5 quintillion bytes of data created daily at our current pace (Marr, 2018). This is divided in to broad categories including: the Internet (searches on Google surpass 40,000 every second), Social Media (users view 4M plus videos each minute), Communication (156M emails sent every minute), Digital Photos (4.7 trillion stored), Services (18M forecast request per minute from the Weather Channel and 600 Wikipedia new page edits per minute by users), and the Internet of Things is expected to add 200B devices by 2020 (Marr, 2018).

An argument could be made that some of the data just described is not valuable, relevant, or useful. But organizations that continue to survive, even flourish, seem to find ways to preserve and use their legacy data. We now share one solution to this problem of accessing and using legacy data.

Purpose of this Research

Demonstrating a middleware solution to access and use legacy data is the purpose of this research. Middleware, including the API and wrapper, became a popular solution to unlock business value (Thiran, Risch, Costilla, Henrard, Kabisch, Petrini, Hainaut, 2005) by exhuming legacy data from aging and sometimes antiquated systems. Persistence (Thomas, 2008) has also emerged, as a viable tool in the hands of programmers who need to unearth data secrets that otherwise would remain buried with maturing software. Users require and expect access to mountains of data right now; this is partially driving the need to reach into legacy systems and provide insight via the smart device in the palm of their hand.

All three of these solutions (API, wrapper, and persistence) find their genesis in dynamic programming languages, but come at a cost with additional runtime checking required (Paulson, 2007) since more instructions must be evaluated at runtime, a fact that is probably moot with the realization of new computing platforms (cloud and SaaS) made possible with Next Generation IT (Thomas, 2008).

The Tiobe Index (Paulson 2007) indicates a significant rise in the use of dynamic languages at the time of the referenced report; further comparison to the June 2019 Tiobe Index (Tiobe, 2019) indicates the use of dynamic languages is still 50% of the top 20 most popular languages in the world, with Python, the most popular dynamic programming language, showing "an all time high in the Tiobe index of 8.5%" with more growth expected in the future. This is significant since dynamic languages are used to create middleware needed to extract legacy data from older information systems.

The other nine dynamic languages in the top 20 June 2019 Tiobe Index report have a sum total popularity of 14.8%. In the middle of this list of 9 is Ruby on Rails with a score of 1.388%.

Using Ruby on Rails for Middleware

As a platform to create middleware Ruby on Rails (RoR) is distinctively suited with a framework following the Model View Controller (MVC) design pattern (Scharlau, 2007). By definition, Rails is a framework built on Ruby, allowing programmers to develop database-focused websites with scaffolding and code generation (Meenakshi, 2015).

In addition to use of the MVC architecture, RoR uses the Create, Read, Update, Delete routing engine to interact with web pages and follows the concept of DRY: "Don't repeat yourself" (Meenakshi, 2015). RoR is taught at many universities as an upper level class to teach skills required to build dynamic websites as part of the computer science curriculum. Specifically, we will explain how it can also be used to build middleware and a system to improve scheduling of classes in a complex environment.

The Environment

Our private university hosts about 3,000 students, half from international locations, who pursue bachelor degrees in the sciences, arts and letters, and professional programs such as business and accounting. Specifically, the authors offer majors in computer science, information technology, and information systems.

Over 20 years ago, a colleague, who is now retired, created an online student management system in Tcl (pronounced Tickle) that allowed academic advisors, faculty, and students to manage and plan a student's academic program. This system mapped a student's classes to complete a major, general education, and minors and is named MAPPER. This ability to "map" or plan the future is especially valuable and sets MAPPER apart from the ERP system (PeopleSoft) used by the registrar, which does not "map" the student's future classes. Additionally, MAPPER allows academic advisors to document appeals, notes, and guidance to students. Grades are also recorded in this system and transfer credits documented.

The development and feature improvement of MAPPER occurred over decades and was continual, based on input from users, primarily student advisors and faculty. Ironically, other systems used to schedule classes did not improve; spreadsheets are still the norm among many academic faculties, departments, and colleges. This may be evident in the fact that 22% of universities practice "just in time" (JIT) scheduling, planning their next term only one academic term in advance (Hanover, 2018).

Scheduling Systems

Research studies show that scheduling is one of the most important and demanding factors impacting student retention at universities (Hanover, 2018). With imperfect tools classes can inadvertently get scheduled at times that interfere with core classes or additional required classes, such as labs. As curriculums and class offerings become more varied and complex the likelihood of conflicts increase. Add to that limited classroom space and multi-use, or specialty (cyber-security sandbox lab, science labs) or high-demand classrooms the scheduling challenge becomes a multifarious problem.

A review of several scheduling systems, including UniTime and Mimosa Scheduling Software revealed very capable systems (Ngoc, 2015) but they did not have the ability to import conflict matrix data from MAPPER. Therefore, a custom development was necessary.

Scheduling System Challenges

The Higher Education Scheduling Index (HESI) annual report of 157 institutions, including four-year private, four-year public, and community colleges discovered that classroom utilization is 67% and seat utilization is 62% even though institutions expressed they felt they were out of space (Ad Astra, 2016). Balancing course access and campus efficiency is a challenge and requirement for a class scheduling system when 36% of entry-level courses are packed with enrollment at 95% in public institutions (Smith, 2016).

Using the Conflict Matrix

The conflict matrix created by MAPPER shows the classes planned for a semester and the conflicts by class for students planning to take the classes. The interpretation of the conflict matrix is done by selecting a class, see Table 1 in the appendices, for example: select CIS 205, the numbers below the asterisk (*) show the number of students in the classes on the lines below that are also MAPPED (planning) to take these other classes. Therefore, of the students planning to take CIS 205, there are 5 also planning to take CS 203, and 6 in CIS 205 are planning to take IS 350. Additionally, the conflict matrix indicates the number of students mapped for a class; for example, CIS 205 has 30 students, shown as (30), followed by a simple code displaying the semesters the class is offered, FWS means Fall, Winter, and Spring, then the name of the class.

Scheduling Classes Pre-Automation

Before Class Scheduler, using the conflict matrix from MAPPER was a manual operation. The seven CS, IS, and IT faculty members would query MAPPER for a current Conflict Matrix and plan a semester with 28 classes on a white board, this process would take about 2 hours. Colleagues, program leads and department chairs at the same university employ various methods to schedule classes, including spreadsheets, white boards, and floating sticky notes. As the champions of teaching automation to increase efficiency we felt the need to practice what we preach and abandon the white board for an automated solution (Fox, 2012).

Requirements for the Class Scheduler

After teaching RoR as an upper-level class for CS students, an idea was born to develop an automated scheduling system, a drag-and-drop online interface that would allow a user to select classes, class locations, times, days, and instructors. The system would also allow the user to color code the different instructor objects.

A requirement for the new system to import conflict matrix data from MAPPER was necessary; additionally, the new system should display conflicts as classes are dropped on a time slot. Conflicts would need to be clearly displayed, showing the number of students planning to take both classes. Simply moving the class object to another time slot or offering two sections of the class could remove the conflict.

2. METHODOLOGY

The Class Scheduler system idea was created while scheduling our classes. Instead of using a whiteboard, and erasing and adding classes to time slots, we thought it would be more efficient to have a digital application with a drag-and-drop interface so we could easily plan a semester of classes. We needed to follow the elements of Agile development to satisfy the needs of the customers (our department) with constant feedback, accept requirement changes on the go at any stage of development, provide constant feedback to our customers, and finally, test the system as each new feature was coded (Hneif, 2009).

Creating Version 1 in Java

A simple Java application was developed and used during our next scheduling meeting. But we still had two problems. First, we needed to look up scheduling conflicts in MAPPER manually and make sure our students could take all of their required classes without conflict. Second, our application did not persist the schedule into a document or database that could easily be shared with the members of our department.

Feature Implementation with RoR

By creating a web application with Ruby on Rails, we provided access to all members of our department and delivered a system that provided productivity with extensive reuse of software (Fox, 2012).

As development continued, it followed a lonely version of Agile, coined Agile Solo (Nyström, 2011), and developed by Watts S. Humphrey in 1993, he used the phrase Personal Software Process (PSP). In this process, a single developer follows an iterative process of planning, development, and postmortem. Development included several steps: requirement, design, coding, and testing. Although a single developer followed the development process in our case, the other members of the department were included as users in the planning and requirements steps.

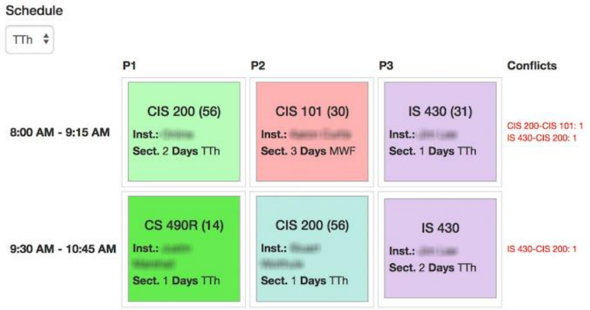
The next version of the Class Scheduler was developed to persist the data into a database and to also include the scheduling conflict data from the legacy MAPPER system. The database has semesters, instructors, courses, terms, and periods (days and times). Using Ruby on Rails, we developed the class schedule so that we can create, update and delete all of these entities. An additional entity, called an offering, is able to connect to a semester and course to an instructor, room, and period through database relationships.

For each semester, we create an offering by dragging a course from a list of all courses, and dropping it into a list of offerings (from left to right, Appendices, Figure 4). From there, we assign the course to an instructor, which color codes the course so we can easily see what each instructor is teaching when looking at a semester schedule.

We then move to a scheduling screen that lays out a blank schedule matrix with rooms along the top as columns and time slots as rows. On the left side of the matrix is a list of courses that still need to be scheduled. By dragging and dropping courses into time slots, we schedule classes. Figure 5 in the Appendices displays the Class Scheduler view that allows the adding of periods (class meeting times).

To process the conflict data from the legacy MAPPER system and to store this data, we create another entity called a conflict. Each conflict is connected to a semester and two course offerings. It also contains the number of students that want to take both courses that semester.

As each class gets scheduled, Class Scheduler queries the conflict table to see if any two classes scheduled for the same time period have any conflicts. We display the conflicts on the right side of the scheduling matrix, see Figure 1. In this case, part of an actual semester schedule, all the conflicts are minimal, since a CIS 200 is offered at two different time slots and also offered as an online course as well (not shown).



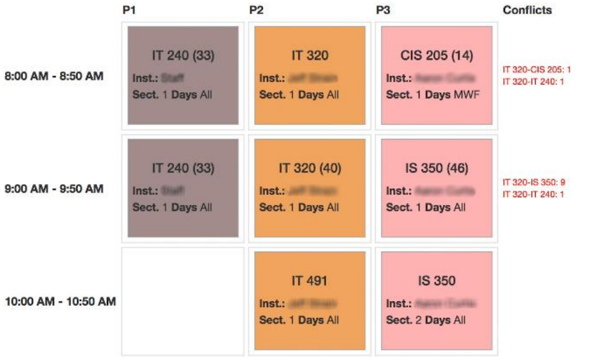
Schedule

TTh

	P1	P2	P3	Conflicts
8:00 AM - 9:15 AM	CIS 200 (56) Inst.: [redacted] Sect. 2 Days TTh	CIS 101 (30) Inst.: [redacted] Sect. 3 Days MWF	IS 430 (31) Inst.: [redacted] Sect. 1 Days TTh	CIS 200-CIS 101: 1 IS 430-CIS 200: 1
9:30 AM - 10:45 AM	CS 490R (14) Inst.: [redacted] Sect. 1 Days TTh	CIS 200 (56) Inst.: [redacted] Sect. 1 Days TTh	IS 430 Inst.: [redacted] Sect. 2 Days TTh	IS 430-CIS 200: 1

Figure 1, T/TH class conflicts, Fall 2019

"CIS200-CIS101:1" is interpreted as these two classes have one conflict. Figure 2 shows a more complex set of conflicts, with "IT 320-IS 350: 9" indicating that 9 students have a conflict if these two classes are scheduled at the same time. These conflicts were avoided by scheduling multiple sections of CIS 350. In both Figure 1 and Figure 2 the conflicts were resolved with multiple sections, but this is not always possible, most of our classes, 13 of 20, have single sections. In Figure 3 in the appendices, there are no conflicts between classes with one section, the Class Scheduler showed such conflicts during the scheduling process and classes were moved around until there were no conflicts between classes with one section.



	P1	P2	P3	Conflicts
8:00 AM - 8:50 AM	IT 240 (33) Inst.: [redacted] Sect. 1 Days All	IT 320 Inst.: [redacted] Sect. 1 Days All	CIS 205 (14) Inst.: [redacted] Sect. 1 Days MWF	IT 320-CIS 205: 1 IT 320-IT 240: 1
9:00 AM - 9:50 AM	IT 240 (33) Inst.: [redacted] Sect. 1 Days All	IT 320 (40) Inst.: [redacted] Sect. 1 Days All	IS 350 (46) Inst.: [redacted] Sect. 1 Days All	IT 320-IS 350: 9 IT 320-IT 240: 1
10:00 AM - 10:50 AM		IT 491 Inst.: [redacted] Sect. 1 Days All	IS 350 Inst.: [redacted] Sect. 2 Days All	

Figure 2, MWF class conflicts, Fall 2019

Implementing Conflict Data

The Class Scheduler needs a way to import the conflicts stored in the legacy MAPPER system. To do this, we copy and paste the conflict matrix from MAPPER into a text area within the Class Scheduler. In Figure 6, we show actual conflict data for 100 conflict records. The Class Scheduler parses the conflict matrix and collects the number of conflicts between each pair of courses. The Class Scheduler matches the courses from MAPPER to courses stored in its own database by name. The conflict import is done after offerings are made for each semester and before classes are scheduled. Figure 3 in the appendices shows a completed semester schedule, which has employed the conflict data for that semester. As a note, a key to the success of this system is the fact that our academic advisors work diligently to make sure student maps are up to date and contain current class schedules.

3.SUMMARY AND CONCLUSIONS

Creating a system that exemplified the principles taught in a RoR class increased class scheduling accuracy. Additionally, the time required to schedule was reduced significantly, most recently, 3 semesters (29 classes, 29 classes, and 17 classes) were scheduled in less then 2 hours.

Planned future research and development activities include:

1. Improve the Class Scheduler system to automatically import conflicts by "screen scrapping" MAPPER's output.

2. Improve the user interface to allow for more rooms to be scheduled, this will allow other departments on campus to use Class Scheduler.

3. Add the ability for Class Scheduler to automatically schedule classes in ERP systems.

4. Research further the state of legacy systems to quantify and describe the extent of data sheltered in legacy systems.

Acknowledgements
We recognize and thank Dr. Don Colton for his innovation and creativity on the MAPPER information system, and especially thank him for his mentorship and leadership. We also thank Dr. Geoff Draper for his initial creation of Class Scheduler, the impetus for this and many other wonderful ideas.

4.REFERENCES

Ad Astra Information Systems (2016, 2018). The Higher Education Scheduling Index. Retrieved 10 June 2019 from https://thehighereducation.com/

Fox, A., & Patterson, D. (2012). Crossing the Software Education Chasm. Communications of the ACM, 55(5), 44-49.

Hanover Research (2018). Best Practices in Course Scheduling. Retrieved 10 June 2019 from https://www.hanoverresearch.com/

Hneif, M., & Ow, S. (2009). Review of Agile Aethodologies in Software Development. International Journal of Research and Reviews in Applied Sciences, 1(1), 1-8.

Marr, B. (2018). How much data do we create every day? The mind-blowing stats everyone should read. Retrieved June 12, 2019 from https://www.forbes.com/

Martens, A., Book, M., & Gruhn, V. (2018). A Data Decomposition Method for Stepwise Migration of Complex Legacy Data. In IEEE ICSE-SEIP '18 Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice, 33-42.

Meenakshi, S. (2015). Ruby on Rails – An Agile Developer's Framework. International Journal of Computer Applications, 112(1).

Ngoc, V. (2015). Teaching and Learning Scheduler System. Vietnam National University, Hanoi, University of Engineering and Technology, Hanoi.

Nyström, A. (2011). Agile Solo-Defining and Evaluating an Agile Software Development Process for a Single Software Developer. A Nyström - 2011 - publications.lib.chalmers.se

Paulson, L. D. (2007). Developers Shift to Dynamic Programming Languages. Computer, 40(2), 12-15.

Powner, D. (2016). Information Technology, Federal Agencies Need to Address Aging Legacy Systems. Testimony Before the Committee on Oversight and Government Reform, House of Representatives. Retrieved 10 June 2019 from http://www.gao.gov/

Scharlau, B. (2007). Teaching Ruby on Rails. Conference Proceedings of the Java and Internet in Computing Curriculum Conference in London Metropolitan University. 24 - 29.

Smith, A. (2016). Fixing Capacity With Better Class Scheduling. Retrieved 10 June 2019 from https://insidehighered.com/

Thiran, P., Risch, T., Costilla, C., Henrard, J., Kabisch, T., Petrini, J., ... & Hainaut, J. L. (2005). Report on the Workshop on Wrapper Techniques for Legacy Data Systems. SIGMOD Record, 34(3), 85-86.

Thomas, D. (2008). Enabling application agility: Software as a Service, Cloud Computing and Dynamic Languages. Journal of object technology, 7(4), 29-32.

Tiobe (2019). Tiobe Index for June 2019. Retrieved June 12, 2019 from https://www.tiobe.com/tiobe-index/

Editor's Note:

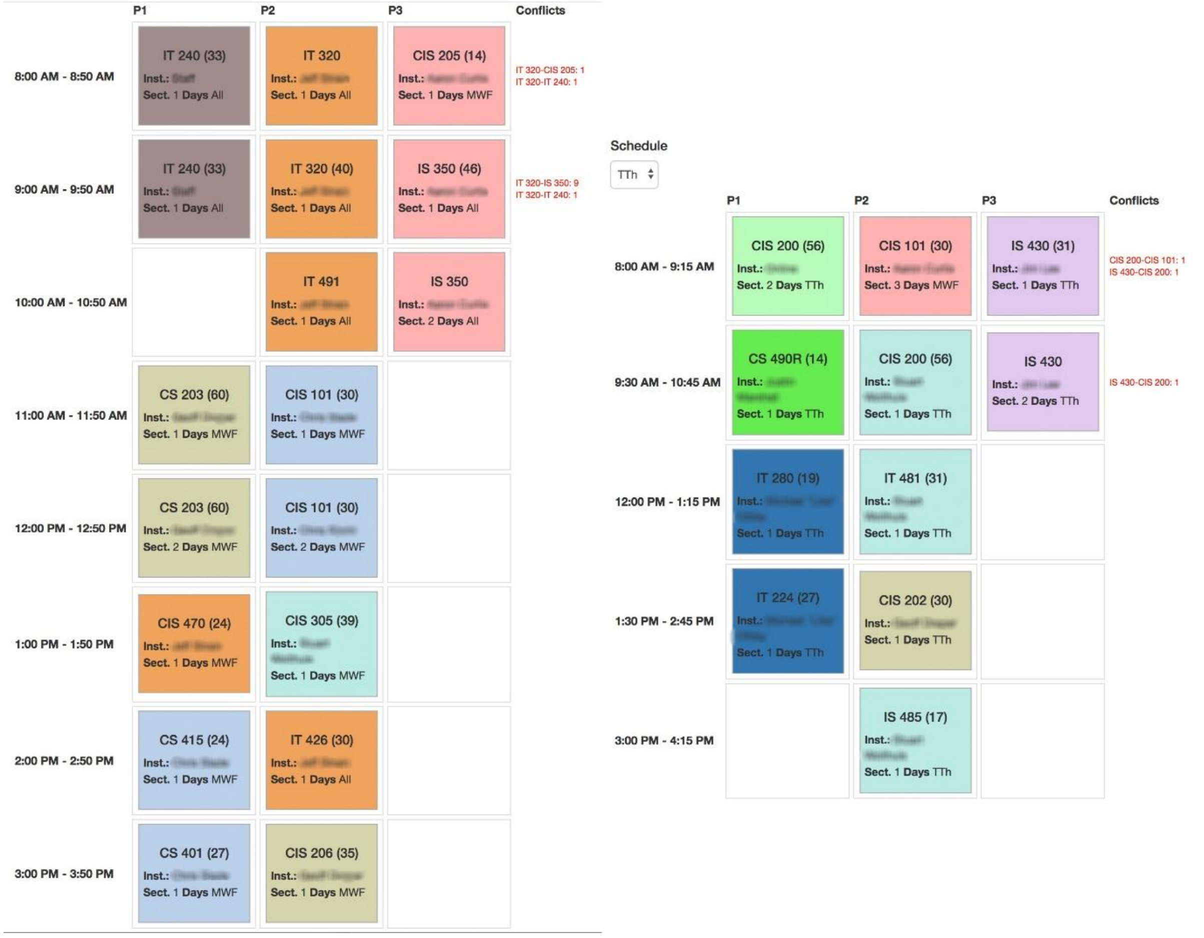
This paper was selected for inclusion in the journal as an EDSIGCON 2019 Meritorious Paper. The acceptance rate is typically 15% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2019.

Appendices

```
*=CIS 101 (101) (FWS) Beginning Programming
8 *=CIS 200 (80) (FWS) Fundamentals of Information Systems and Technology
2 6 *=CIS 202 (46) (FWS) Object-Oriented Programming I
1 1 + *=CIS 205 (30) (FWS) Discrete Mathematics I
1 1 . 2 *=CIS 206 (38) (F) Discrete Mathematics II
1 . 2 3 + *=CIS 305 (45) (FW) Systems Engineering I
. . . . 1 . *=CIS 470 (22) (FS) Ethics in Computer and Information Sciences
1 4 . 5 + + 1 *=CS 203 (60) (F) Object-Oriented Programming II
. . . . 1 4 8 1 *=CS 401 (29) (F) Web Applications Development
. . . . 1 4 8 1 + *=CS 415 (26) (F) Operating Systems Design
. . . . . 1 5 . + + *=CS 490R (18) (FWS) Advanced Topics in Computer Science
2 3 4 6 + + . + . 1 2 *=IS 350 (62) (FW) Database Management Systems
. 3 . . 4 3 4 4 2 . . 4 *=IS 430 (36) (FW) Foundations in IT Services, Enterprise Systems, and ERP Skills
. . . . 3 . 4 . . . . . 5 *=IS 485 (18) (FS) Project Management and Practice
5 3 . 1 1 1 1 1 . 1 1 3 1 . *=IT 224 (37) (FWS) Computer Hardware and Systems Software
6 2 . 1 1 1 . 1 . . . 4 . . + *=IT 224L (25) (FWS) Computer Hardware and Systems Software Lab
+ 8 6 . . 1 1 3 . 1 1 1 . . 6 4 *=IT 240 (57) (FWS) Fundamentals of Web Design and Technology
7 5 7 4 3 1 . 4 1 1 . 3 . . 7 5 4 *=IT 280 (32) (FWS) Computer Networking
3 3 5 4 . + 2 1 2 2 1 + 1 3 5 5 1 5 *=IT 320 (45) (F) Linux Essentials
. . 1 . 3 6 + 1 2 . . 2 8 + . . . . + *=IT 426 (35) (F) Computer Network Services

. means zero, + means 10 or more
```

Table 1: Conflict Matrix for All Students (2195) Fall 2019

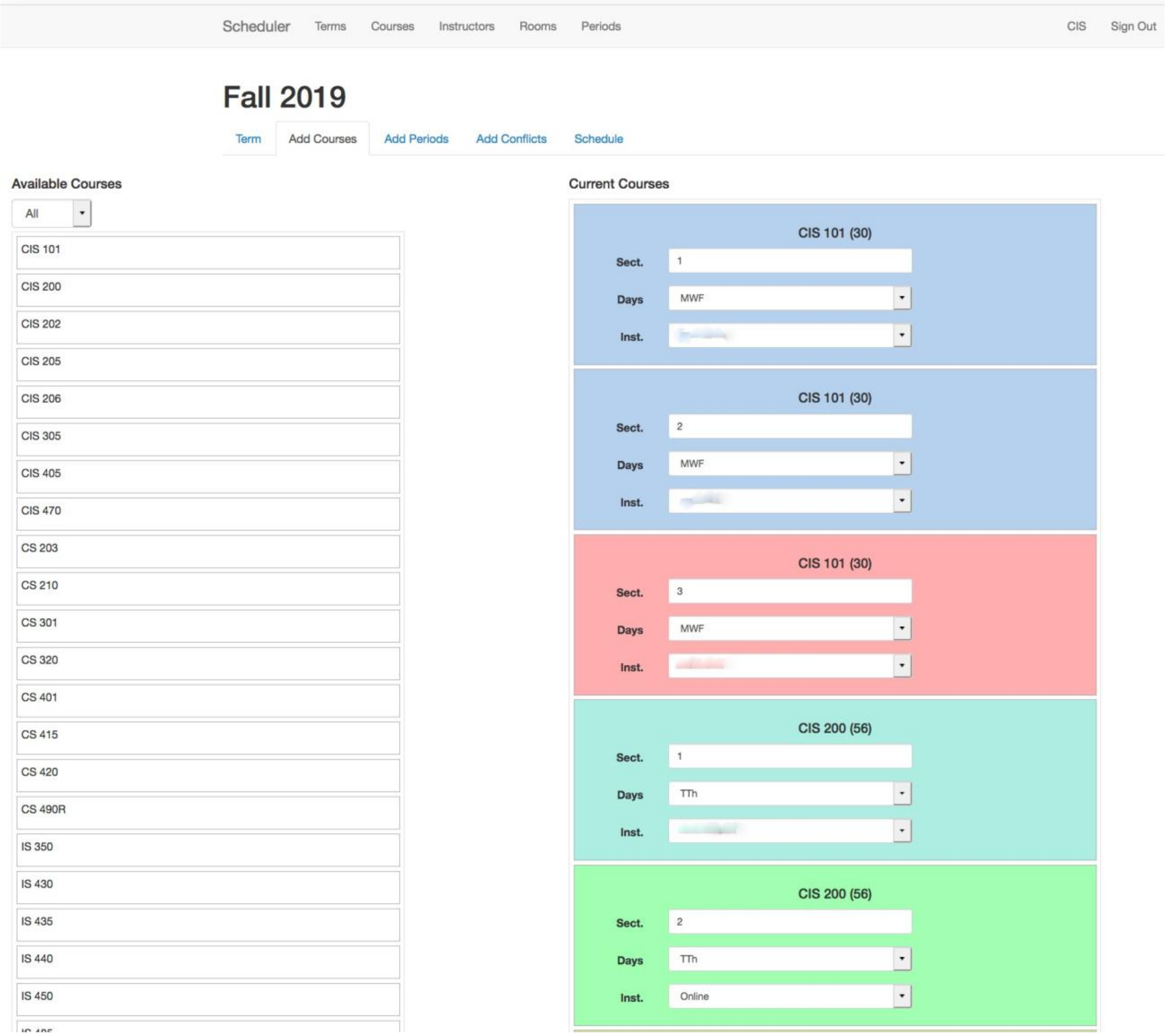


	P1	P2	P3	Conflicts
8:00 AM - 8:50 AM	IT 240 (33) Inst.: [redacted] Sect. 1 Days All	IT 320 Inst.: [redacted] Sect. 1 Days All	CIS 205 (14) Inst.: [redacted] Sect. 1 Days MWF	IT 320-CIS 205: 1 IT 320-IT 240: 1
9:00 AM - 9:50 AM	IT 240 (33) Inst.: [redacted] Sect. 1 Days All	IT 320 (40) Inst.: [redacted] Sect. 1 Days All	IS 350 (46) Inst.: [redacted] Sect. 1 Days All	IT 320-IS 350: 9 IT 320-IT 240: 1
10:00 AM - 10:50 AM		IT 491 Inst.: [redacted] Sect. 1 Days All	IS 350 Inst.: [redacted] Sect. 2 Days All	
11:00 AM - 11:50 AM	CS 203 (60) Inst.: [redacted] Sect. 1 Days MWF	CIS 101 (30) Inst.: [redacted] Sect. 1 Days MWF		
12:00 PM - 12:50 PM	CS 203 (60) Inst.: [redacted] Sect. 2 Days MWF	CIS 101 (30) Inst.: [redacted] Sect. 2 Days MWF		
1:00 PM - 1:50 PM	CIS 470 (24) Inst.: [redacted] Sect. 1 Days MWF	CIS 305 (39) Inst.: [redacted] Sect. 1 Days MWF		
2:00 PM - 2:50 PM	CS 415 (24) Inst.: [redacted] Sect. 1 Days MWF	IT 426 (30) Inst.: [redacted] Sect. 1 Days All		
3:00 PM - 3:50 PM	CS 401 (27) Inst.: [redacted] Sect. 1 Days MWF	CIS 206 (35) Inst.: [redacted] Sect. 1 Days MWF		

Schedule: TTh

	P1	P2	P3	Conflicts
8:00 AM - 9:15 AM	CIS 200 (56) Inst.: [redacted] Sect. 2 Days TTh	CIS 101 (30) Inst.: [redacted] Sect. 3 Days MWF	IS 430 (31) Inst.: [redacted] Sect. 1 Days TTh	CIS 200-CIS 101: 1 IS 430-CIS 200: 1
9:30 AM - 10:45 AM	CS 490R (14) Inst.: [redacted] Sect. 1 Days TTh	CIS 200 (56) Inst.: [redacted] Sect. 1 Days TTh	IS 430 Inst.: [redacted] Sect. 2 Days TTh	IS 430-CIS 200: 1
12:00 PM - 1:15 PM	IT 280 (19) Inst.: [redacted] Sect. 1 Days TTh	IT 481 (31) Inst.: [redacted] Sect. 1 Days TTh		
1:30 PM - 2:45 PM	IT 224 (27) Inst.: [redacted] Sect. 1 Days TTh	CIS 202 (30) Inst.: [redacted] Sect. 1 Days TTh		
3:00 PM - 4:15 PM		IS 485 (17) Inst.: [redacted] Sect. 1 Days TTh		

Figure 3: Class Scheduler, Completed Semester from Schedule View



Fall 2019

Available Courses

All

- CIS 101
- CIS 200
- CIS 202
- CIS 205
- CIS 206
- CIS 305
- CIS 405
- CIS 470
- CS 203
- CS 210
- CS 301
- CS 320
- CS 401
- CS 415
- CS 420
- CS 490R
- IS 350
- IS 430
- IS 435
- IS 440
- IS 450

Current Courses

CIS 101 (30)

Sect.	1
Days	MWF
Inst.	[redacted]

CIS 101 (30)

Sect.	2
Days	MWF
Inst.	[redacted]

CIS 101 (30)

Sect.	3
Days	MWF
Inst.	[redacted]

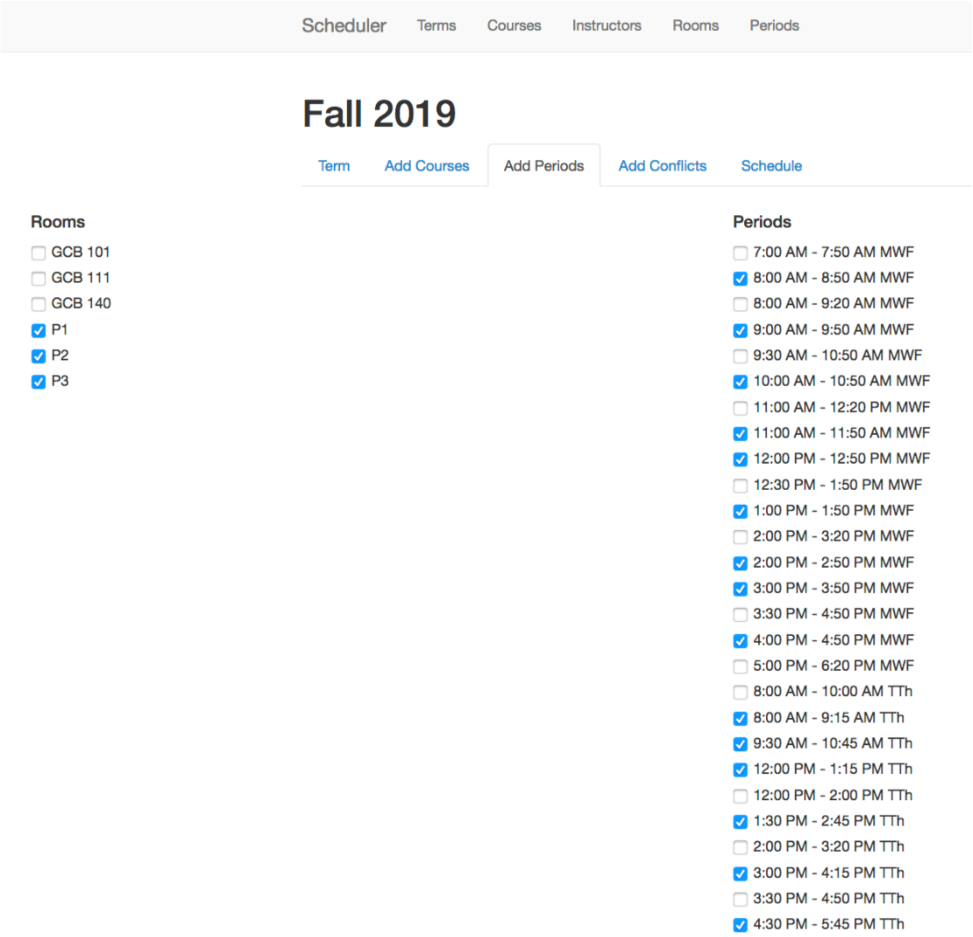
CIS 200 (56)

Sect.	1
Days	TTh
Inst.	[redacted]

CIS 200 (56)

Sect.	2
Days	TTh
Inst.	Online

Figure 4: Class Scheduler Add Courses View



Fall 2019

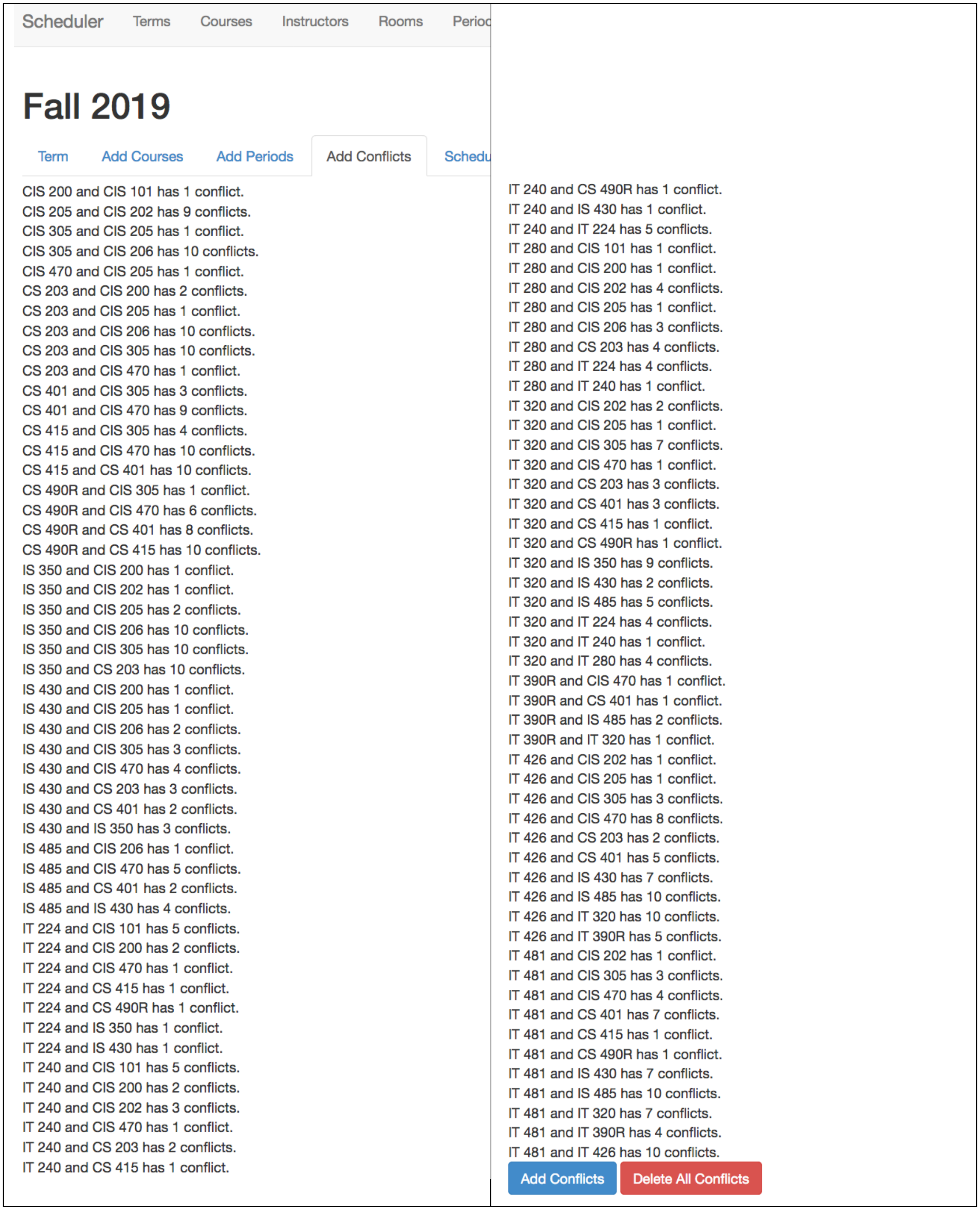
Rooms

□ GCB 101
□ GCB 111
□ GCB 140
☑ P1
☑ P2
☑ P3

Periods

□ 7:00 AM - 7:50 AM MWF
☑ 8:00 AM - 8:50 AM MWF
□ 8:00 AM - 9:20 AM MWF
☑ 9:00 AM - 9:50 AM MWF
□ 9:30 AM - 10:50 AM MWF
☑ 10:00 AM - 10:50 AM MWF
□ 11:00 AM - 12:20 PM MWF
☑ 11:00 AM - 11:50 AM MWF
☑ 12:00 PM - 12:50 PM MWF
□ 12:30 PM - 1:50 PM MWF
☑ 1:00 PM - 1:50 PM MWF
□ 2:00 PM - 3:20 PM MWF
☑ 2:00 PM - 2:50 PM MWF
☑ 3:00 PM - 3:50 PM MWF
□ 3:30 PM - 4:50 PM MWF
☑ 4:00 PM - 4:50 PM MWF
□ 5:00 PM - 6:20 PM MWF
□ 8:00 AM - 10:00 AM TTh
☑ 8:00 AM - 9:15 AM TTh
☑ 9:30 AM - 10:45 AM TTh
☑ 12:00 PM - 1:15 PM TTh
□ 12:00 PM - 2:00 PM TTh
☑ 1:30 PM - 2:45 PM TTh
□ 2:00 PM - 3:20 PM TTh
☑ 3:00 PM - 4:15 PM TTh
□ 3:30 PM - 4:50 PM TTh
☑ 4:30 PM - 5:45 PM TTh

Figure 5: Class Scheduler Add Periods View



Fall 2019

```
CIS 200 and CIS 101 has 1 conflict.
CIS 205 and CIS 202 has 9 conflicts.
CIS 305 and CIS 205 has 1 conflict.
CIS 305 and CIS 206 has 10 conflicts.
CIS 470 and CIS 205 has 1 conflict.
CS 203 and CIS 200 has 2 conflicts.
CS 203 and CIS 205 has 1 conflict.
CS 203 and CIS 206 has 10 conflicts.
CS 203 and CIS 305 has 10 conflicts.
CS 203 and CIS 470 has 1 conflict.
CS 401 and CIS 305 has 3 conflicts.
CS 401 and CIS 470 has 9 conflicts.
CS 415 and CIS 305 has 4 conflicts.
CS 415 and CIS 470 has 10 conflicts.
CS 415 and CS 401 has 10 conflicts.
CS 490R and CIS 305 has 1 conflict.
CS 490R and CIS 470 has 6 conflicts.
CS 490R and CS 401 has 8 conflicts.
CS 490R and CS 415 has 10 conflicts.
IS 350 and CIS 200 has 1 conflict.
IS 350 and CIS 202 has 1 conflict.
IS 350 and CIS 205 has 2 conflicts.
IS 350 and CIS 206 has 10 conflicts.
IS 350 and CIS 305 has 10 conflicts.
IS 350 and CS 203 has 10 conflicts.
IS 430 and CIS 200 has 1 conflict.
IS 430 and CIS 205 has 1 conflict.
IS 430 and CIS 206 has 2 conflicts.
IS 430 and CIS 305 has 3 conflicts.
IS 430 and CIS 470 has 4 conflicts.
IS 430 and CS 203 has 3 conflicts.
IS 430 and CS 401 has 2 conflicts.
IS 430 and IS 350 has 3 conflicts.
IS 485 and CIS 206 has 1 conflict.
IS 485 and CIS 470 has 5 conflicts.
IS 485 and CS 401 has 2 conflicts.
IS 485 and IS 430 has 4 conflicts.
IT 224 and CIS 101 has 5 conflicts.
IT 224 and CIS 200 has 2 conflicts.
IT 224 and CIS 470 has 1 conflict.
IT 224 and CS 415 has 1 conflict.
IT 224 and CS 490R has 1 conflict.
IT 224 and IS 350 has 1 conflict.
IT 224 and IS 430 has 1 conflict.
IT 240 and CIS 101 has 5 conflicts.
IT 240 and CIS 200 has 2 conflicts.
IT 240 and CIS 202 has 3 conflicts.
IT 240 and CIS 470 has 1 conflict.
IT 240 and CS 203 has 2 conflicts.
IT 240 and CS 415 has 1 conflict.
IT 240 and CS 490R has 1 conflict.
IT 240 and IS 430 has 1 conflict.
IT 240 and IT 224 has 5 conflicts.
IT 280 and CIS 101 has 1 conflict.
IT 280 and CIS 200 has 1 conflict.
IT 280 and CIS 202 has 4 conflicts.
IT 280 and CIS 205 has 1 conflict.
IT 280 and CIS 206 has 3 conflicts.
IT 280 and CS 203 has 4 conflicts.
IT 280 and IT 224 has 4 conflicts.
IT 280 and IT 240 has 1 conflict.
IT 320 and CIS 202 has 2 conflicts.
IT 320 and CIS 205 has 1 conflict.
IT 320 and CIS 305 has 7 conflicts.
IT 320 and CIS 470 has 1 conflict.
IT 320 and CS 203 has 3 conflicts.
IT 320 and CS 401 has 3 conflicts.
IT 320 and CS 415 has 1 conflict.
IT 320 and CS 490R has 1 conflict.
IT 320 and IS 350 has 9 conflicts.
IT 320 and IS 430 has 2 conflicts.
IT 320 and IS 485 has 5 conflicts.
IT 320 and IT 224 has 4 conflicts.
IT 320 and IT 240 has 1 conflict.
IT 320 and IT 280 has 4 conflicts.
IT 390R and CIS 470 has 1 conflict.
IT 390R and CS 401 has 1 conflict.
IT 390R and IS 485 has 2 conflicts.
IT 390R and IT 320 has 1 conflict.
IT 426 and CIS 202 has 1 conflict.
IT 426 and CIS 205 has 1 conflict.
IT 426 and CIS 305 has 3 conflicts.
IT 426 and CIS 470 has 8 conflicts.
IT 426 and CS 203 has 2 conflicts.
IT 426 and CS 401 has 5 conflicts.
IT 426 and IS 430 has 7 conflicts.
IT 426 and IS 485 has 10 conflicts.
IT 426 and IT 320 has 10 conflicts.
IT 426 and IT 390R has 5 conflicts.
IT 481 and CIS 202 has 1 conflict.
IT 481 and CIS 305 has 3 conflicts.
IT 481 and CIS 470 has 4 conflicts.
IT 481 and CS 401 has 7 conflicts.
IT 481 and CS 415 has 1 conflict.
IT 481 and CS 490R has 1 conflict.
IT 481 and IS 430 has 7 conflicts.
IT 481 and IS 485 has 10 conflicts.
IT 481 and IT 320 has 7 conflicts.
IT 481 and IT 390R has 4 conflicts.
IT 481 and IT 426 has 10 conflicts.
```

Figure 6: Class Scheduler Add Conflicts View

Undergraduate Business Analytics and the overlap with Information Systems Programs

Wendy Ceccucci
wendy.ceccucci@qu.edu

Kiku Jones
kiku.jones@qu.edu

Katarzyna Toskin
katarzyna.toskin@qu.edu

Computer Information Systems
Quinnipiac University
Hamden, CT

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

Abstract

As companies continue to put data and business analytics as their top priority, universities will need to supply students with the appropriate skill sets that meet this demand and offer future opportunities to their graduates. Although business analytics is a new field, many of the required competencies stem from already established areas such as Information/Computer Technology or Information Systems. Using a sample of 225 randomly selected AACSB accredited business schools this study examined the new developments in Business Analytics undergraduate academic programs, and determined the amount of overlap between the Business Analytics and the Information/Computer Technology degree programs. Our findings reveal that approximately 36 percent of the Business Analytics programs overlap with the Information/Computer Technology programs. In addition, the top three required courses in most Business Analytics programs include a Database course, predictive analytics course, and Introduction to Business Analytics. This research provides valuable insight for schools that haven't adopted a Business Analytics degree yet or are looking to improve their existing curriculum. In addition, colleges and universities can now utilize the appropriate Information Systems courses and include them as important foundation and part of their Business Analytics programs.

Keywords: Data Analytics, Business Analytics, Business Intelligence, Business Analytics program,

1. INTRODUCTION

Holsapple et al. (2014) defined Business Analytics as "evidence-based problem recognition and solving that happen within the context of business situations" (p. 134). Although business analytics is a fairly new term, it originates from the decision support systems that were introduced in the late 1960s, followed by Business Intelligence systems in the late 1980s (Watson, 2011). Many of these traditional techniques, however, used and analyzed structured data to support their business decisions. The evolution of the Internet in 1970s and wide adoption of the World Wide Web, mobile devices, as well as sensor technology, have allowed companies to generate and collect more data than ever before (Chen et al. 2012). Furthermore, this data comes in a new format such as audio, video, text, which is no longer structured. Therefore, the availability of ubiquitous and unstructured data, has created a demand for novel techniques and data analysis skills. As stated by Holsapple et al. (2014) "modern-day BA [Business Analytics] is rooted in the ongoing advances of systems to support decision making. These advances include increasingly powerful mechanisms for acquiring, generating, assimilating, selecting, and emitting knowledge relevant to making decisions."

Subsequently the skills required for today's business decision making have evolved as well. Chiang et al. (2012) identified three categories of skills required for effective Business Analytics professionals, presented in Figure 1:

1. Analytical skills such as statistical analyses, data and text mining, optimization and simulation.
2. Information Technology (IT) Knowledge and Skills including relational databases, data warehousing, visualization and dashboard design, semi structured and unstructured data management and manipulation, and more.
3. Business Knowledge and Communication Skills focusing on business domain knowledge and ability to propose analytical solutions as well as articulate findings.

From the origin of business analytics, the definition, as well as the skills model, it is evident that there is an overlap between the analytical skills and Information Technology. More specifically, Information Technology knowledge and skills appear to play a fundamental role in developing Business Analytics professionals.

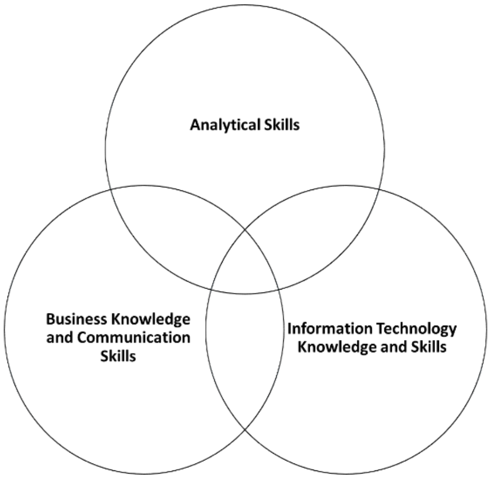

Analytical Skills
Business Knowledge and Communication Skills
Information Technology Knowledge and Skills

***Figure 1*. Business Analytics Skills Model**

The Demand for Business Analytics Majors
Students with a degree in Business Analytics have a variety of different job opportunities. A few of the careers for Business Analytics graduates include: Business Analyst, Quantitative Analyst, Market Research Analyst, Financial Analyst, and Operations Research Analyst. The demand for professional in business analytics is increasing. (Univ Wash, 2018)

By 2022, 85% of companies are expected to adopt data analytics (Bytyci, 2019). Careers in business and data analytics are in demand right now (Labbe, 2018). Therefore, more graduates with data and business analytic skills are needed. Business analytics utilizes data (collecting, storing, and analyzing) from business intelligence and customers in order to generate plans for business enhancements in efficiency and revenue (Gorman and Klimberg, 2014; Wilder and Ozgur, 2015). Traditionally business analysts focused on core business, with knowledge in all business principles, and primarily sought to understand and develop requirements for an information system project. Now business analysts are focused on collecting, storing, and analyzing data they "help guide businesses in improving processes, products, services and software through data analysis. …. workers straddle the line between IT and the business to help bridge the gap and improve efficiency" (pg 1, Pratt & White, 2019). According to IBM, by 2020 business analyst positions will increase from 364,000 to 2,720,000 openings (Arora, 2018; Bytyci, 2019), as companies have revealed the need for analysis to determine valuable business insights. With the increasing demand for better business and customer knowledge, organizations are increasingly relying on business analytics.

While some business roles are set to decline in demand because of technology, such as auditors and banking clerks, business analyst jobs will advance with technology. Therefore, there could be more overlap between information system degrees and business analytics degrees. In 2015-16, Cleary (2019) examined business analysis job postings. She found five top degree types employers requested for entry-level business analysts – general administration and management, computer science, finance, information systems, and accounting. Further analysis and discussions found that business analysts need a mix of both information technology and business skills, which has been echoed by other authors (Noodle Editorial Staff, 2019). Gorman and Klimberg (2014) found business analytics to combine statistics with information systems as well as quantitative methods. However, traditional business schools are struggling to produce graduates that can effectively meet the growing industry demands (LeClaire, 2016). Many business schools are offering master degrees in business analytics or data analytics (UNC Institute for Advanced Analytics, 2019; Labbe, 2018), but these schools are slower to offer programs at the undergraduate level.

The purpose of this paper is to examine the offerings of business analytics or data analytics majors and minors at the undergraduate, business school level to determine what, if any, overlap might exist with the information system programs. Findings of this study will provide important insight for universities that are looking to start a Business Analytics program or want to strengthen their existing BA curriculum. More specifically this study will identify which courses converge between the two fields of Business Analytics and Information Systems, helping schools create balanced curriculum for a Business/Data Analytics program that is and will continue to be in high demand for years to come.

2. RESEARCH METHODOLOGY

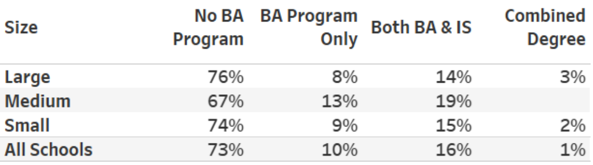
Data was collected from 225 randomly selected AACSB accredited business schools. This accounts for approximately 39.8% of the total AACSB business schools. Data was collected from the schools' catalog and material available on their website. The schools were first searched to see if they had an analytics program within the school of business. Seventy-four programs (33%) had some type of business analytics program, either major, minor, certificate or concentration. For two of the schools, the program requirements could not be determined from their websites and the schools were removed from further analysis. Due to varying types of BA requirements within schools, the BA programs were classified as having a "program" and worth further investigation if a school required 15 credits (5 classes) or more. There was a total of 58 schools that were considered as offering BA programs. Each school that had a qualified BA program was then further evaluated to see if the school offered an IS program. Data was collected on the courses for both Business Analytics program and the corresponding IS program.

A classification of courses needed to be created for the data collected to be mapped. Appendices A and B show the list for required courses and elective courses. Once the data was collected, data mapping began. This study followed a similar process used in a prior study which surveyed university IS program curriculum described on their websites (Yang & Wen, 2017). Multiple authors went through the courses and coded them based on the classification item that best described each particular item. If a course did not fit in any of the classifications, a new classification was added to the list and communicated to the other coder. After the authors went through the list and coded the courses, the lists were reconciled against each other. Where there were differences or questions, the authors discussed those until a decision was made.

3. FINDINGS AND DISCUSSION

Of all the schools evaluated, 34 of the programs offered degrees in both IS and Analytics and 3 offered combined degrees. To further clarify, if the program was a specialization, minor or a concentration that required more than 5 courses or 15 credits it was considered as offering a program in BA.

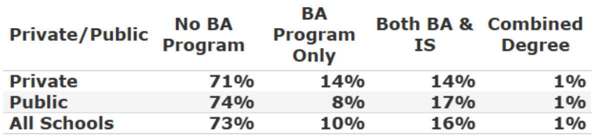
The size of the schools based upon undergraduate enrollment and the programs that are offered are given in Table 1. Small universities were categorized as small if undergraduate enrollment was less than 10,000 students. Medium size schools had enrollment between 10,000 and less than 20,000. Large schools had enrollment greater than or equal to 20,000. The size of the school had no effect on whether the school offered a BA program. Seventy-three percent of all the schools did not offer any BA program and 16% offered both a BA and IS program.



Size	No BA Program	BA Program Only	Both BA & IS	Combined Degree
Large	76%	8%	14%	3%
Medium	67%	13%	19%	
Small	74%	9%	15%	2%
All Schools	73%	10%	16%	1%

Table 1. University Size and Program Offerings

Approximately 66% of the schools analyzed were public schools (Table 2). Fourteen percent of the private schools offered both, BA and IS programs and 17% of the public schools offered both.



Private/Public	No BA Program	BA Program Only	Both BA & IS	Combined Degree
Private	71%	14%	14%	1%
Public	74%	8%	17%	1%
All Schools	73%	10%	16%	1%

Table 2. University Type and Program Offerings

Business Analytics Course Requirements

Figure 2 shows the math requirements for the BA programs. The percentage indicated in orange are those schools that the math class is required as part of the college or school's core requirements. Fifty-six percent of the schools require at least one statistic course and 45% require at least one calculus course.

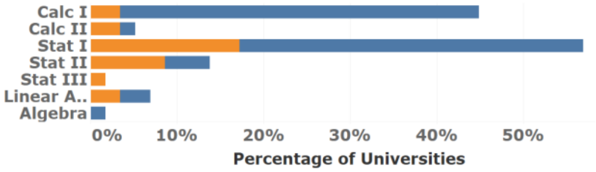


Figure 2. BA Math Requirements

Figure 3 shows the programming requirements for the BA programs. Programming courses are not typically part of the core but an introductory programming course is required by 36% of the BA programs. Twelve percent require a statistical programming course.

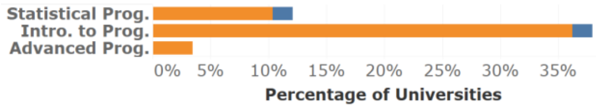


Figure 3. BA Programming Requirements

Appendix A shows the different required courses offered by the BA programs. A course in database management is the most frequently required course and is required by 67% of all BA programs. Fifty-six of the classifications were found to be required courses in at least one of the BA programs. The top 20% of the required courses, excluding courses in the business or university core are found in Table 3. This large list of courses shows that there is no agreed upon set of skills that should be obtained when completing a BA program. Without that consistency, it is difficult to know which concepts a student has been exposed to without knowing the program he/she completed. For example, thirty-three percent of the programs do not require a database course. However, this does not necessarily mean those students are not being exposed to these concepts. Perhaps database skills are being taught in another course being offered. The Intro to BA course may be a course which covers these skills.

Course	Univ. Count	Percentage
Database	39	67%
Predictive Analytics 1	29	50%
Intro to BA	28	48%
Dec. Models 1	23	40%
Intro to Programming	21	36%
Capstone	16	28%
Predictive Analytics II	14	24%
Data Viz	14	24%
Data Warehousing	13	22%
Big Data	11	19%
Stat I	10	17%

Table 3. BA Required Courses

Business Analytics Course Electives

Appendix B lists all the different electives offered by the BA programs. Some schools have statistics as an elective and not a required course. The list shows the vast array of different offerings among programs. Among the elective list are many of the courses that are considered required courses for other BA programs. This further illustrates the fact that there is still no single set of agreed upon courses in a BA program. Electives are where a BA program can easily incorporate other disciplines. Courses in areas such as Sports Analytics, Healthcare Analytics, Supply Chain/Logistics, Human Resource Management Data, etc., can be developed with faculty from those areas. Or perhaps, these courses already existed prior to the creation of the BA program? This can help to tap into a new group of students interested in analytics for that particular industry/area.

BA and IS Overlap

As previously stated, there were 34 programs that offered both BA and IS programs. In looking at just these schools, it appears that many of the required BA courses are also required in the CIS programs or as part of the core business program. Appendix C. looks at those schools that have both a CIS and a BA program and shows the overlap of the required BA courses and how it relates to CIS and the school core. For example, the database course is the most frequently required course in Business Analytics Programs. Out of the 35 schools that offer both CIS and BA programs, 26 of the BA programs require a database course. Out of these 26, 6 schools do not have it as either a CIS elective or a CIS required course; 18 schools have the database listed as a required course in the CIS program and 2 schools have the database course listed as a CIS elective.

Looking at the predictive analytics course, of the 35 schools that offer BA and CIS programs, 19 of the school's BA programs require a predictive analytics course. Out of these 19 schools, 13 do not require the predictive analytics course for the CIS major. One school requires the predictive analytics course for both the CIS and BA major. And 5 schools allow the predictive analytics course to count the predictive analytics course as an elective. Of note, of the top 20% required BA courses, only three were not in at least 20% of the IS programs in the universities reviewed. These three courses were Big Data, Data Warehousing and Capstone.

How much is the overlap? Table 4 shows the percentage of required BA Classes that are also required CIS courses. For example, looking at university id 411 80% of the required BA courses are also required for the CIS major and 33% of all courses required and electives can also be taken for the CIS major.

As industry demands continue to grow with more business analytics positions, well-informed prospective students will be looking for universities that prepare students for these careers. While many universities have begun to build BA programs at the master's level, undergraduate programs in BA are also needed. This research indicates that many schools, with a current IS program, may already have the courses in place to begin a BA program. As stated,
it was determined that having five courses (15 credits) was the marker for having a BA program. For many universities, these courses may be found within the IS program itself or in other departments, such as economics or math. This can make it easier to begin the process of starting the BA program.

Univ. ID	Overlap Required BA & CIS	Overlap of all BA with CIS
411	80.00%	33.30%
378	66.70%	85.70%
128	50.00%	35.70%
360	50.00%	71.40%
439	50.00%	50.00%
446	50.00%	100.00%
92	42.90%	42.90%
377	37.50%	68.80%
109	33.30%	33.30%
215	33.30%	33.30%
367	33.30%	14.30%
32	30.00%	63.60%
218	25.00%	10.00%
9	20.00%	22.20%
224	16.70%	50.00%
550	16.70%	58.30%
80	15.40%	15.40%
387	14.30%	14.30%
124	12.50%	45.00%
13	0.00%	28.60%
25	0.00%	35.30%
161	0.00%	20.00%
188	0.00%	0.00%
201	0.00%	0.00%
242	0.00%	50.00%
252	0.00%	0.00%
271	0.00%	28.60%
273	0.00%	20.00%
278	0.00%	13.00%
338	0.00%	0.00%
340	0.00%	66.70%
423	0.00%	50.00%
485	0.00%	10.00%
531	0.00%	42.90%
Overall Average	19.90%	35.70%

Table 4. Percentage of BA Classes that are also required CIS Courses

The information found in this study can be used by universities to review the courses already offered by their IS program. While there is no one set curriculum followed for a BA program, this Top 20% required list provides guidance as to what is considered important across many of the universities reviewed in this sample. This will allow university IS programs to determine which courses need to be added and perhaps which courses need to be adapted in order to properly offer a BA program. For example, they may wish to change the introduction to programming course to utilize R as the language taught to satisfy both IS and BA programs.

4. CONCLUSION

According to PWC, data scientists, data engineers and business analysts are among the most sought-after positions in America (PWC, 2018). As demand continues to rise for employees with business analytics skills, universities will need to find ways to prepare their students for these opportunities.

This study found that there is quite a bit of overlap in BA and IS programs. Universities who have not yet begun to build a BA program should consider first looking at their existing courses. They may find that they only need to develop a couple of courses to complete the BA program offering. This would be a good time to consider setting the program apart from others by specializing in a particular area, e.g., healthcare.

This study illustrates that there is no clear curriculum for a basic undergraduate BA program. At this time, no expectation can be made in regard to skills acquired when a student graduates with such a program. This is something that should be considered as the field develops. Future research should propose a model curriculum.

5. REFERENCES

Arora, V. (2018). "What are the expected business analytics professionals demand in 2019?" *BA Times,* Retrieved from https://www.batimes.com/articles/what-are-the-expected-business-analytics-professionals-demand-in-2019.html

Bichler, M., Heinzl, A. & van der Aalst, W.M.P. (2017) Business Analytics and Data Science: Once Again?, *Business Information Systems Engineering 59*(2):77–79.

Bytyci, R. (2019). "The future of business analysis," *BA Times*, Retrieved from https://www.batimes.com/articles/the-future-of-business-analysis.html.

Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS Quarterly, 36*(4).

Chiang, R. H., Goes, P., & Stohr, E. A. (2012). Business intelligence and analytics education, and program development: A unique opportunity for the information systems discipline. ACM Transactions on Management Information Systems (TMIS), 3(3), 12.

Cleary, J. "Career focus: what employers want from a business analyst," Retrieved from https://mbs.rutgers.edu/articles/career-focus-what-employers-want-business-analyst

Demchenko, Y., Belloum, A., Los, W., Wiktorski, T., Manieri, A., Brocks, H., & Brewer, S. (2016). EDISON data science framework: a foundation for building data science profession for research and industry. In 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom) (pp. 620-626). IEEE.

Gorman, M.F. & Klimberg, R.K. (2014). "Benchmarking academic programs in business analytics," *INFORMS Journal on Applied Analytics, 44*(3), 329-341.

Holsapple C, Lee-Post A, & Pakath R (2014). A unified foundation for business analytics. *Decision Support Systems, 64*, 130–141.

Labbe, M.(2018) "Rising demand for business analytics programs," *TechTarget*, Retrieved fromhttps://searchbusinessanalytics.techtarget.com/feature/Rising-demand-for-business-analytics-education-programs.

LeClair, D. (2016). "Big data's big future in business education." *AACSB Blog.* Retrieved from http://www.aacsb.edu/blog/2016/march/big-data-big-future-in-businesseducation?utm_source=HighRoads%20Solutions&utm_medium=Email&utm_campaign=HighRoads%20All%20Emails

Noodle Editorial Staff (2018) "A 2018 guide for a career in business analytics," Retrieved from https://www.noodle.com/articles/business-analytics-career-paths

Pratt, M. & White,S. (April, 2019) "What is a business analyst? A key role for business – IT efficiency" *CIO.* Retrieved from { https://www.cio.com/article/2436638/project-management-what-do-business-analysts-actually-do-for-software-implementation-projects.html

PWC (2018) "What's next for the data science and analytics job market?" Retrieved from https://www.pwc.com/us/en/library/data-science-and-analytics.html

UNC Institute for Advanced Analytics (2019) "Graduate Degree Programs in Analytics and Data Science,", Retrieved from https://analytics.ncsu.edu/?page_id=4184

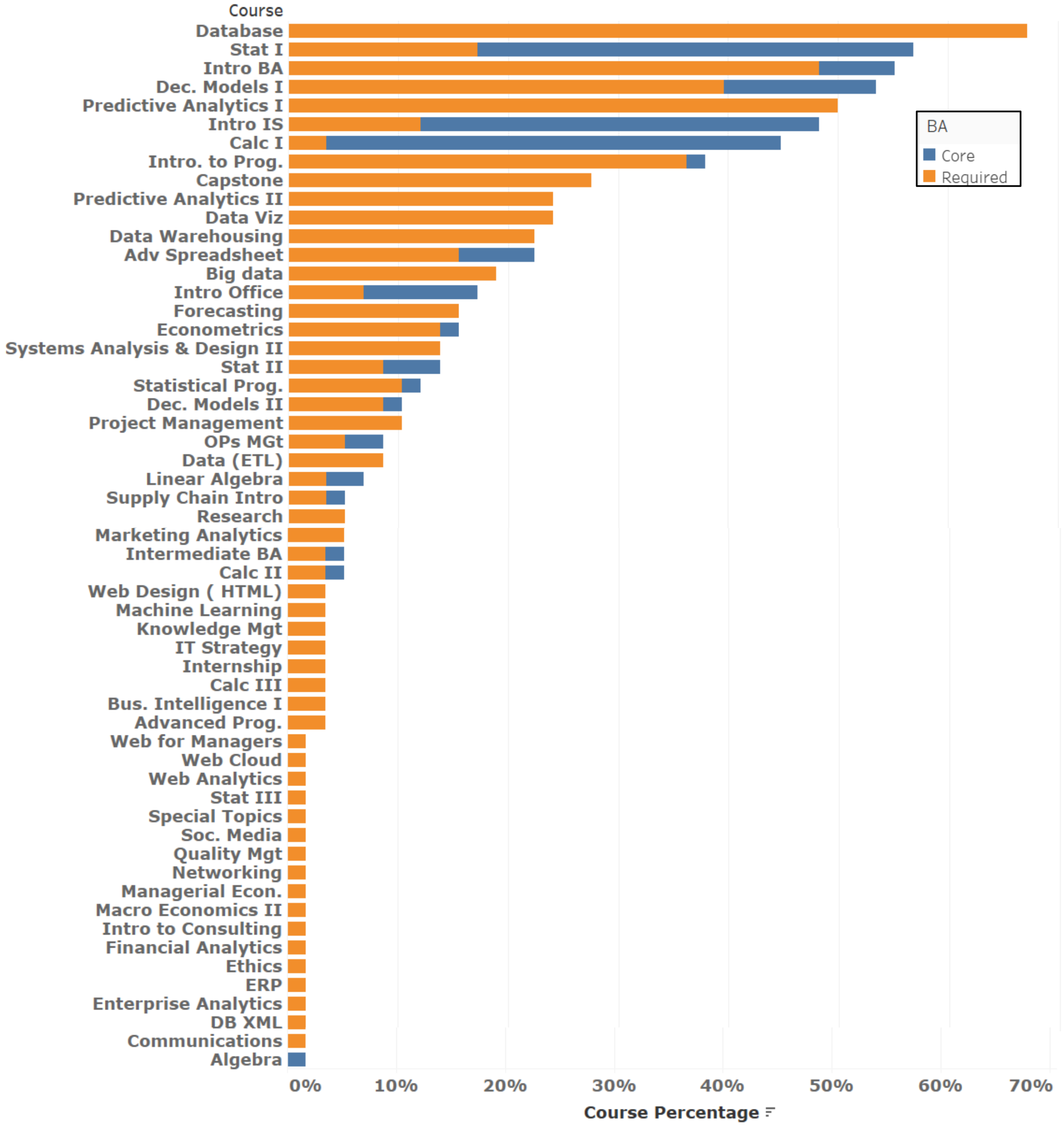
University of Washington Foster Blog (2018). "What jobs can I get with a business analytics degree", Retrieved from https://blog.foster.uw.edu/jobs-can-get-business-analytics-degree/

Watson, H. (2011) Business analytics insight: hype or here to stay? *Business Intelligence Journal, 16*(1), 4–8.

Wilder, C.R. & Ozgur, C.O. (2015). "Business analytics curriculum for undergraduate majors," *INFORMS Transactions on Education, 15*(2), 180-187.

Yang, S. C., & Weng, B. (2017). "Toward a cybersecurity curriculum model for undergraduate business schools: A survey of AACSB-accredited institutions in the United States." *Journal of Education for Business, 92*(1), 1-8.

Appendix A Business Analytics Required Courses

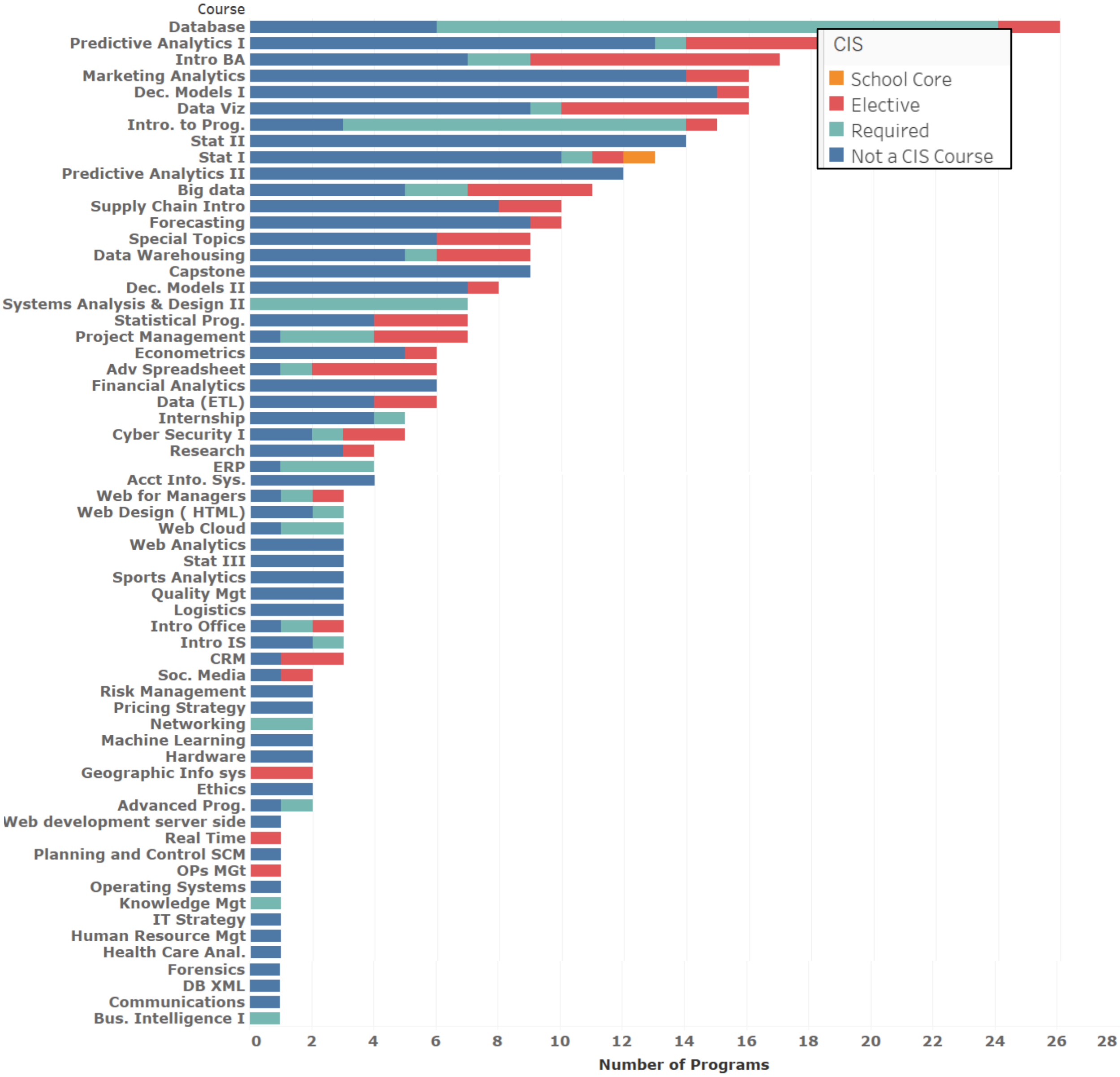


Appendix B. Business Analytics Elective Courses

Course	*Percentage*	*Number*
Marketing Analytics	40%	23
Special Topics	29%	17
Stat II	22%	13
Financial Analytics	22%	13
Supply Chain Intro	21%	12
Acct Info. Sys.	21%	12
Intro. to Prog.	19%	11
Data Viz	19%	11
Project Management	17%	10
Dec. Models II	17%	10
Dec. Models I	17%	10
Database	17%	10
Forecasting	16%	9
Econometrics	16%	9
Systems Analysis & Design II	14%	8
Stat I	14%	8
Statistical Prog.	12%	7
Predictive Analytics II	12%	7
Internship	10%	6
Logistics	9%	5
Intro BA	9%	5
Cyber Security I	9%	5
Soc. Media	7%	4
Predictive Analytics I	7%	4
Health Care Anal.	7%	4
ERP	7%	4
Data Warehousing	7%	4
CRM	7%	4
Web for Managers	5%	3
Web Design (HTML)	5%	3
Web Cloud	5%	3
Web Analytics	5%	3
Sports Analytics	5%	3
Quality Mgt	5%	3
Pricing Strategy	5%	3
Stat III	3%	2
Risk Management	3%	2
Research	3%	2
Real Time	3%	2
Planning and Control SCM	3%	2
Networking	3%	2
Managerial Econ.	3%	2

Knowledge Mgt	3%	2
Intro IS	3%	2
Human Resource Mgt	3%	2
Hardware	3%	2
Geographic Info sys	3%	2
Game Theory	3%	2
Energy Analytics	3%	2
Data (ETL)	3%	2
Bus. Intelligence I	3%	2
Big data	3%	2
Web development server side	2%	1
Systems Analysis & Design I	2%	1
Real Estate	2%	1
Operating Systems	2%	1
Negotiations	2%	1
Machine Learning	2%	1
Leadership	2%	1
Info. Storage and Mgt	2%	1
Graphics	2%	1
Forensics	2%	1
Ethics	2%	1
Dec. Models III	2%	1
DB XML	2%	1
Contract Theory	2%	1
Advanced Prog.	2%	1
Adv Spreadsheet 2%	2%	1

Appendix C. Overlap of BA and CIS courses



Lizards in the Street! Introducing Cybersecurity Awareness in a Digital Literacy Context

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

Birgy Lorenz
birgy.lorenz@ttu.ee
School of Information Technologies
Tallinn University of Technology
Tallinn, Estonia

Abstract

Learning cybersecurity awareness builds on basic information technology concepts and digital literacy skills. In an effort to raise cybersecurity awareness among information technology students, this paper describes a series of three different interactive sessions offered to students of all levels at a business university. The sessions introduced cybersecurity awareness through identifying actual breaches and incidents, using open source intelligence tools, and participating in a capture the flag style competition. Student comments in blog posts and interviews after these sessions show the relevance of cybersecurity awareness in their daily lives and a general sense of surprise, amazement and concern at how much personal information is readily available online.

Keywords: cybersecurity awareness, digital literacy, open source intelligence tools, hacking competition

1. INTRODUCTION

Skills and competencies related to cybersecurity awareness are often included as part of courses in digital, computer, or information literacy, or as elements of life-long learning. (Ala-Mutka, Punie, & Redecker, 2008; *AP Computer Science Principles*, 2017; Chinien & Boutin, 2011) These "21st Century Skills"(van Laar, van Deursen, van Dijk, & de Haan, 2017) are vital at home, at the workplace and to function in society. Despite the perception that today's digital natives (Prensky, 2012) are tech savvy and have been born with a security mindset, having a baseline set of knowledge, skills, and abilities can go a long way toward developing core cybersecurity competencies common to many work roles (Dawson & Thomson, 2018).

Universities have introduced technical degree programs in cybersecurity to meet industry demand for graduates with specialized skills. Some courses include in-class exercises using online tools to provide hands-on experience of technical concepts such as virtualization and infrastructure automation (Marquardson, 2018), and performance testing in an isolated environment (Marquardson & Gomillion, 2018).

Cybersecurity awareness related skills often are much more applied, focusing on competencies such as good password management (using different secure passwords, storing passwords safely using a password manager, two-factor authentication), recognizing phishing attempts, detecting malicious emails, and using open source intelligence (OSINT) tools. Combining intuition, curiosity and the ability to search and analyze data gathered from the Internet and other open sources is a powerful skill to detect fake news, scams, and social manipulation in the world. (Bada, Sasse, & Nurse, 2019; Wells, Conflict, & Gibson, 2017)

Digcomp, a digital competence framework for European citizens, (Carretero-Gomez, Vuorikari, & Punie, 2017) presents competencies to protect devices and personal data from risks and threats in digital environments, and applies cybersecurity skills to realistic employment scenarios, such as the use of social media in a corporate environment. While the United States National Cyber Strategy ("National-Cyber-Strategy.pdf," 2018) points out the need to protect networks, services and information, and secure critical infrastructure, Despite all of the technology precautions in place in the workplace, organizations are realizing that humans are still the weakest link in cybersecurity (Boulton, 2017; Postimees, 2019; Zimmermann & Renaud, 2019). As an example, one recent study found that most novice users do not know how to encrypt their email messages.(Ruoti et al., 2016)

"Some say that the average computer user simply lacks knowledge and awareness of cybersecurity issues and of the secure behaviors they ought to be carrying out… [and] other researchers argue that users do not care about possible consequences, [and] are unmotivated to take responsibility." (Zimmermann & Renaud, 2019, p. 4)

2. CYBERSECURITY AWARENESS AND DIGITAL LITERACY

Cybersecurity awareness relies on individuals knowing basic ways that they can protect themselves, their data and their devices. The foundation of that awareness may be found in developing basic technology and digital literacy skills.

Digital Literacy Skills

Digital literacy skills have evolved from gaining proficiency with productivity tools, email, the World Wide Web, social media, collaboration tools, mobile devices and the cloud (Dijk & Deursen, 2014; Frydenberg & Press, 2010) to creating, organizing, sharing, and reusing online content, accessing information across devices and platforms, and maintaining privacy and identity online. (Wheeler, 2010)

When learning about cybersecurity, introductory IT courses often cover the importance of communicating safely online, demonstrating the use of computers safely and responsibly, making judgment about digital content when evaluating and repurposing it for a given audience, demonstrating responsible use of online services; selecting, combining, and using Internet services; understanding the potential of information technology for collaboration when computers are networked; using online services securely; recognizing that persistence of data on the Internet requires careful protection of online identity; understanding ethical issues surrounding the application of information technology. (*AP Computer Science Principles*, 2017; Harris & Patten, 2015) These digital literacy skills are crucial for mastering cybersecurity awareness.

Cybersecurity Skills

Stenmap (Mäses, Randmann, Maennel, & Lorenz, 2018) is a model to classify cybersecurity-related skills. Competencies range from non-cybersecurity specific to cybersecurity-specific skills along the horizontal axis, and non-technical to technical skills along the vertical axis.

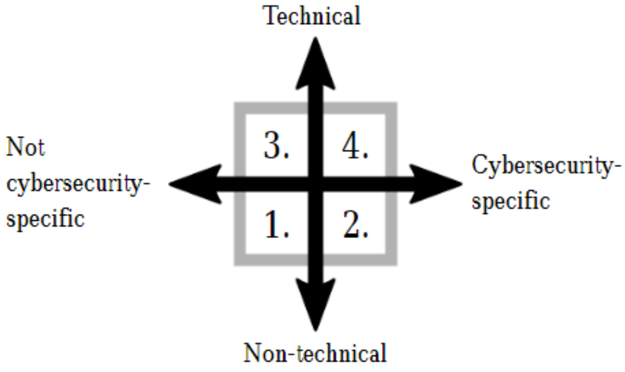


Figure 1. Classifying Cybersecurity Skills

Quadrant 1 includes skills that are non-technical and not cybersecurity-specific, such as leadership and communication skills. Team and group exercises require these highly valued skills. Quadrant 2 includes skills that are cybersecurity-specific, but non-technical, such as identifying phishing emails or the importance of secure passwords. Quadrant 3 includes technical skills that may not be cybersecurity related, such as coding and basic understanding of browsers or the Internet. Quadrant 4 requires skills that are both technical and cybersecurity-specific, such as implementing encryption or an SQL injection attack.

Mäses notes that "it is not always easy to position a skill in this Cybersec-Tech window. For example, skills related to reporting could be general nontechnical or very specific and technical. Nevertheless, this Cybersec-Tech window can help to facilitate a discussion about which skills a cybersecurity exercise should target."(Mäses et al., 2018, p. 9)

Figure 2 adapts Figure 1, listing specific digital literacy skills and where they fall within the Stenmap model:

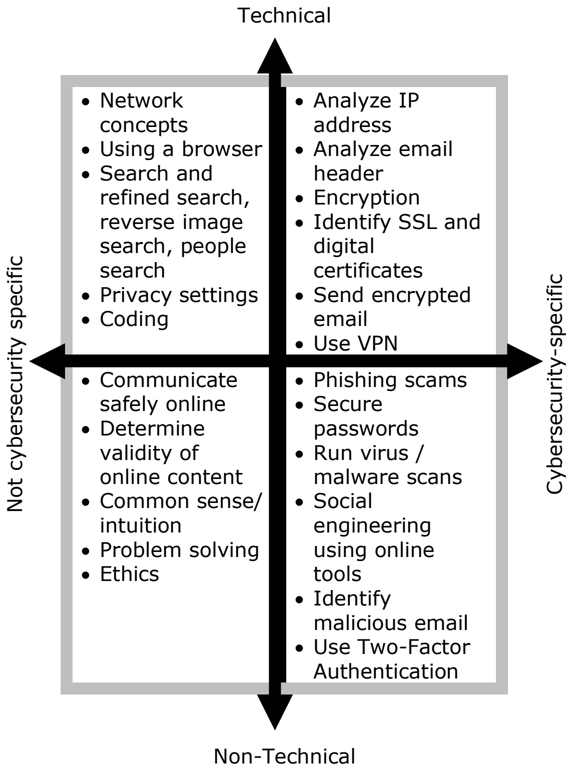

Technical
• Network concepts
• Using a browser
• Search and refined search, reverse image search, people search
• Privacy settings
• Coding
• Analyze IP address
• Analyze email header
• Encryption
• Identify SSL and digital certificates
• Send encrypted email
• Use VPN
Not cybersecurity specific
Cybersecurity-specific
• Communicate safely online
• Determine validity of online content
• Common sense/ intuition
• Problem solving
• Ethics
• Phishing scams
• Secure passwords
• Run virus / malware scans
• Social engineering using online tools
• Identify malicious email
• Use Two-Factor Authentication
Non-Technical

Figure 2. Applying Digital Literacy Competencies to Cybersecurity Skill Classifications

The AP College Board, in its computer science principles course, posits that "cybersecurity is an important concern for the Internet and the systems built on it."(*AP Computer Science Principles*, 2017, p. 34) Students should be able to identify existing cybersecurity concerns and potential options to address them. Issues of awareness mentioned include impact of DDoS attacks, hardware, software, and human components of cybersecurity; phishing, viruses, and other attacks; foundations and applications of cryptography; digital certificates. In the AP College Board Computer Science principles course, the focus on cybersecurity awareness is from an Internet-based perspective.

Open Source Intelligence Tools

Open source intelligence (OSINT) tools have emerged as an important components for locating, organizing, and differentiating recognizing new types of relevant information online. (Glassman & Kang, 2012) OSINT information and data include social media sites and online social networks, public records databases, photos, maps, and images, online surveillance cameras, code repositories, media websites. OSINT tools include special purpose search engines and other applications that can quickly gather and analyze data from hundreds of websites, perform fact-checking, scan files for viruses and malware, and determine the technology platforms used on a website. (Kissiah & eInvestigator.com, 2019)

Knowing the appropriate tools makes it possible to perform tasks such as determining which social networks have a given username registered, searching for photos and images to determine their authenticity, evaluating a user's Twitter habits; identifying common patterns in user passwords, encoding messages and files, obtaining information from an IP address search, and analyzing email headers. Knowing about several of these tools is one way to demonstrate cybersecurity awareness and digital literacy skills.

Guiding Questions

Given the importance of raising cybersecurity awareness among students from both technology and general backgrounds, the following guiding questions for this study emerge:

- What concepts, skills, and applications must students know to demonstrate cybersecurity awareness?
- What OSINT tools can students use to prepare for the cybersecurity challenges that they will face?
- How can these be presented in ways that introduce or reinforce digital literacy concepts and skills that students learn in an introductory IT course?

3. METHODOLOGY

To provide outside of class, informal opportunities for students to learn about cybersecurity, the presenter offered three 80-minute interactive sessions on cybersecurity topics, biweekly between February 5 and March 5, 2019. Sessions were open to all students at XXXXXXX University, a business university in XXXXXXXXXXXXX. The topics of these sessions are shown in Table 1:

Table 1. Cybersecurity Awareness Session Topics

Session 1	Cybersecurity Stories
Session 2	Open Source Intelligence Tools and How to Hack through Search
Session 3	Capture the Flag (CTF) Style Hacking Competition

While the three cybersecurity awareness events were not tied to a single course, instructors of introductory IT, web design, database, cybersecurity, and other undergraduate CIS courses encouraged their students to attend. In addition, two IT instructors and two technology administrators on campus attended two of the sessions.

Participants self-selected to attend these events, and used their own devices (laptops, tablets or mobile devices). Some instructors offered extra credit to students in their classes who wrote a short report after attending. An average of 20 participants attended each session, with 24 participants attending the final CTF session. Session 2 on OSINT Tools was recorded, and the video posted online for the benefit of students who were unable to attend, or who wanted to review prior to the competition in Session 3. The study used an action research method (Johnson, 2012) where the presenter was actively participating in the lectures as a facilitator and as the source of cybersecurity facts.

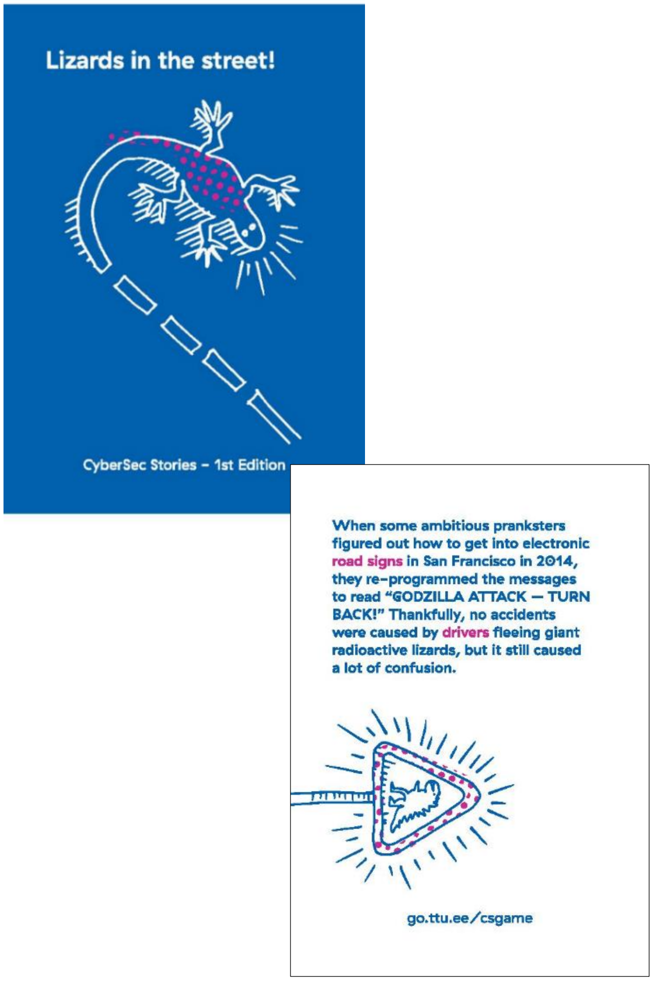
Each session took place in a technology lab where students sat at tables to facilitate group work; the room had two projection screens for participants to see the presenter's slides easily. The first two sessions were methodologically lecture with hands-on practice exercises and the final CTF session was structured as a team competition.

Session 1: Cybersecurity Stories

The first presentation provided a general overview of cybersecurity concepts and cases that happen in Internet realms. The content of the sessions were text, pictures, videos, a game called CyberSec Stories 1 (Lorenz, 2018) and open discussion. CyberSec Stories is a card game focusing on various security cases in the digital world. The game was developed by persons that are involved with Tallinn University of Technology Centre for Digital Forensics and Cyber Security scientists, lecturers, students, and partners. The game consists of 54 cases that help to raise overall awareness of cybersecurity. Players take turns reading a short headline on the card (such as, "Lizards in the street!") and then try to guess what happened. The reverse side of each card contains a short summary of the case for members to read to give clues to their teammates, or the team can search online to find out more information.

A sample game card is shown in Figure 3. "Lizards in the street!" refers to an electronic road sign in San Francisco that was hacked to read "Godzilla Attack! Turn back!" (Rosenblum, 2014) All of the game cards for CyberSec Stories 1st Edition are available at https://sites.google.com/view/tty-csgame/.

Topics include how big is the Internet today and how the hacker mindset works. Cases discussed included how to crash a car with piece of paper; who one becomes professional with just typing spaces; why a digital company might need to force everyone to use paper systems for 6 months; why companies in Ukraine infected their own systems with virus; trusting people because of face value or because they wear a uniform; how to deal with ransomware and what can happen when you answer spam email.



When some ambitious pranksters figured out how to get into electronic road signs in San Francisco in 2014, they re-programmed the messages to read "GODZILLA ATTACK — TURN BACK!" Thankfully, no accidents were caused by drivers fleeing giant radioactive lizards, but it still caused a lot of confusion.

go.ttu.ee/csgame

Figure 3. Front and back of a CyberSec Stories game card.

The presentation concluded with a discussion of minimal cybersecurity skills that students need to function in the world today, and students shared their own cybersecurity stories and experiences.

Session 2: Open Source Intelligence Tools
The second session featured a presentation on Open Source Intelligence (OSINT) tools to find and determine the validity of online information. The presentation included slides, videos, small group exercises, and open discussion. Topics included three different hats of a hacker (white, gray, and black); the term *OSINT*; and several OSINT tools to locate analyze online data.

Appendix I Table 1 contains several OSINT tools, many of which were demonstrated during this session.

Students tried some of the OSINT tools working in small group exercises. Exercises had them create a fake online persona using websites to generate fictitious names, locations, occupations, and profile photos; they completed a phishing quiz; analyzed information available from their IP address, and determined if their personal account information has been compromised in a recent data breach.

Session 3: Capture the Flag Competition
The series concluded with a Capture the Flag (CTF) style competition where participants worked in self-selected teams to solve cybersecurity-related challenges or evaluate truthful information online. The puzzles were of varying difficulty and required participants to exercise different skill sets to solve. Many of the solutions involved using OSINT tools presented in the previous session.

"In the cybersecurity world, 'capture the flag' competitions are the simulated crucible in which the curriculum lessons are tested and validated by the students. Instead of a playing field with physical flags to capture, … teams defend and attack computer networks and the flags are data and services that are either preserved or disabled."(Serapiglia, 2016, p. 28) Some CTF competitions may last for a few hours, a day or more; participants may be students, enthusiasts, or professionals. Players can attempt the various challenges individually, or they can work with team members to attempt to score the highest number of points. Once an individual challenge is solved, a flag, or code value, is given to the player and they submit this flag to the CTF server to earn points.

Exercises included: looking at secret data contained in a file (GPS address, additional text inside the picture); detecting problems such as missing hardware components in a computer; analyzing pictures to find a password; decrypting code or solving puzzles using mobile phone, base64 encoder, and a book; analyzing email headers; and finding an alternative way to access websites that have been geoblocked. (Geoblocking is a means of refusing incoming requests for web content that originate in specific countries.)

Many of the solutions relied on students grasp of digital literacy skills and technology concepts: understanding parts of a URL, recognizing an IP address, using a search engine effectively, evaluating social media posts; using productivity software, and other topics. Students were given hints as needed once the competition was underway.

Sample CTF exercises and puzzles are shown in Appendix 2. Please contact the authors for more information.

4. RESULTS AND DISCUSSION

The authors gathered immediate feedback after each session and feedback within two weeks after the final session by asking students enrolled in an introductory technology concepts course to write a short blog post describing their impressions and lessons learned and what they think college students should know about cybersecurity. The authors also discussed with faculty teaching the Introductory Computing Concepts course about possibilities and challenges to integrate some of these cybersecurity awareness exercises and concepts in the current course.

Sessions 1 and 2 Debrief

Session 1 discussions analysis show the topics raised from the first session: the idea was to talk about actors on the internet, connect history into modern world and inventions and discuss what competencies one should have when finishing their college experience. Usually, in awareness sessions, people tend to talk about social media and passwords, here the talk went to a deeper level - related more on technology and its possibilities. Session 2 topics analysis focused more on OSINT possibilities, also how hackers think, how phishing is done (different techniques to detect hacking, malicious content), developing a fake online persona, using fake pictures and videos, social engineering and ethics.

Exercises chosen for Session 2 were based on applying common digital literacy skills to demonstrate cybersecurity competencies. OSINT exercises were related to finding information from the Internet, such as identifying photos of real and fake Picasso works of art.

Steganography, the practice of concealing information within a message, image, or video file, was used to demonstrate how one might hide information inside a file, analogous to how hackers might hide malicious code in email attachments. Forensics exercises let participants detect phishing and viruses from the email header or hash analyze changes in the server or website; GPS exercises let participants discover how to find out where a picture was taken. Hardware exercises taught about how the computer is made, how the network is built. Cryptography exercises helped participants understand secret codes and language ciphers.

Most worrisome and interesting to participants were discussions about hackers, viruses and how to analyze malicious emails, OSINT and its techniques and social engineering.

Students' and teachers' feedback centered around how to detect problems, gather evidence and get to know all these cases on a deeper level. Discussions around competencies listed the need to have overall awareness and understanding how the Internet works.

Discussions also showed that there has not been a conversation about what kind of security skills should one have when finishing university. Teachers identified links between cybersecurity awareness and critical thinking; students were much more practical in wanting to learn tangible skills such as understanding passwords habits and how to deal with constant flow of emails (spam and phishing attacks), or even whom to turn when something happens. without being ashamed. When completing the hands-on activities, participants wanted to know which OSINT tools and websites to use to solve the exercises.

The sessions also brought up ethical discussions of issues such as: Who is to blame when code is insecure? Who is responsible for the security of personal data stored online?

Session 3: CTF Debrief

In Session 3, the CTF competition, of the 80 minutes available, 10-15 minutes were used to give an introduction and organize groups; 50 minutes were available to complete the activity, and 15-20 minutes at the end were available to debrief. The presenters learned that the time available for the exercise (approximately 50 minutes) was insufficient to complete most of the 25 exercises provided. Students solved most of the easier level OSINT exercises as they were most used to using Google or another search engine to gather answers for homework or personal life needs. For example, exercises had students find the default password for a Wi-Fi router or detect a missing word from a news headline. Students were also successful in completing the visual exercises (such as to find a password from a photo taken in a professor's office). Hardest exercises (most of which were not solved) were related to cryptography, analyzing code form the website or computer screen from server logs. It was interesting that even though the best teams accomplished approximately one-third of the exercises and need to strategize on how to do them, they were so happy that they had used the computer and developed critical thinking skills by solving puzzles, detecting problems and proposing solutions.

Feedback showed that most of the groups (8 groups, 3 people in each) found different exercises that were interesting to them and from what they were empowered the most. A similar theme was that when they worked in a team to help each other rather than working individually, they accomplished more; also solving the hardest exercises on which they spent most of their time were those that impressed them the most. They pointed out various tools and websites they learned about during the session.

Student Comments
After attending at least one of the three sessions, several participants wrote blog posts on "What should the college students know and learn about cybersecurity?" Feedback from student blogs (which were completed within two weeks of the final session) showcased the relevance of cybersecurity awareness in their own lives. The biggest impact topics were how to use search tools, logical filtering and social engineering skills to acquire information about people, places, companies and how to analyze it as a hacker would; and how to analyze data (website, email, personal) legitimacy for updating the defense of being phished.

One student said: "I feel as though many students are unaware of many issues that come along with cybersecurity or lack thereof in this case. Throughout this year, I and many other students, have received countless phishing emails that cause devices to obtain viruses if you click a certain link. Towards the beginning of this year, it was obvious when an email was a scam, however, more recently it seems like they have been disguised a lot better. For example, I received emails that were from my close friends about topics that we both had sent or received emails about. This made me realize that because a friend of mine was hacked, hackers had some of my information as well. An email about a cheer event was sent to me from my teammate's email account and was very believable until I realized the suspicious layout of the email. Overall, I believe it'd be useful to include one class during the IT101 course that is devoted to identifying when an email is unsafe and how to prevent viruses from computers."

Students pointed out a better understanding of how to use safety precautions (need for more complex passwords, contained online presence, evaluated use of media tools) and minimize risks as in the process of exercises they could experience being also in the attacker side. At the end of the sessions, they did an audit of their own devices and environments, and passwords to improve their online safety and experience.

Privacy was a concern for students from the point of view of a consumer and a marketer. One future marketing major suggested:

- ○ Discuss clearing cookies how does this have an impact on marketers? Why or why not should cookies be cleared?
- ○ Discuss privacy in terms of social media advertisements. What do timely, relevant ads mean for the consumer?
- ○ Discuss the legality and ethics behind big data and privacy. Why should there be a federal definition of what big data is?
- ○ Discuss privacy - example how can we tell when a job offer is a scam? Is the offer from social media or sent by email legit?
- ○ Should information like our social security number, financial information of other information be submitted on an application?

Students commented on what they thought they knew about cybersecurity before the session, and the lessons they learned: "Before this class I feel like I had the general knowledge of cybersecurity that comes with growing up in my generation. Certainly, always err on the side of caution and assume non trusted sites and emails are not safe. I did know that you were supposed to change your password frequently and that passwords should be a complex variation of numbers, letters, and symbols. I did not know there were sites that you could run emails and other media through to scan for viruses. I also learned a lot about the variations of different viruses and malware. Aside from viruses and malware this is also a whole section of cybersecurity which directly involves protection from hackers and people. People who use the internet to attack others can do so in a variety of ways. Even social hacking can be implemented to steal information about someone from a third party which you assume would be secure."

Students were taken by the amount of information available through social media posts. Said one student: "It can be surprising how much information that someone can find about you just by looking at old tweets or Instagram posts. College students are already aware of employers looking through social media accounts, but they need to be more aware about what they post because cyberhackers can find anything. I believe that this big lesson in here is, do not post things that you do not want strangers to find out about you."

Said another student: "At the cybersecurity workshop, I learned lots of different methods to approach our computers and personal information. The most important one for me is the email with links. Once people click into the links, their personal information would be taken by hackers. We were taught how to distinguish real or fake emails. Basically, we look at the senders and other information in the email to make sure its authority. And if we click into links or accidentally go into random websites, we do not give out any personal information including bank information. I think that is important because it is close to our life. Other things that people should know is how to protect their all kinds of accounts. Such as how to make sure no one logs in their accounts."

Some also got inspired by the exercises to develop decoding experiences for others, others had more inspired by learning more about ethical hacking overall or history of cybersecurity. A few people also asked about career possibilities in the field. Students also wanted to know how hacking works (from the actions of the hacker, providing demonstrations) and how to recover after being hacked, clicking a bad link, or sharing information that should have remained private.

Students' concerns with cybersecurity also had to do with keeping their phones safe, protecting their social media data, not being taken by phishing scams, and determining the validity of online information.

5. CONCLUSIONS AND FUTURE WORK

Developing cybersecurity awareness skills is crucial for preparing students to take their place as information technology workers in their future careers. The ability to detect spam, phishing, malware, and other attacks, as well as the ability to maintain privacy of one's information online and determine the validity of information online are valuable skills whose foundations require basic digital and technology literacy skills.

This paper presented a model for classifying cybersecurity skills within the context of digital literacy and described three different sessions for raising cybersecurity awareness at a university through an interactive game, open source intelligence tools, and a capture the flag style competition Any of the three sessions can be incorporated into a technology concepts course or shared as an extracurricular activity to raise cybersecurity awareness. Student results suggest that these sessions were informative and increased interest in keeping students' data and devices safe.

In future iterations of this project, the authors will update and present current OSINT tools, describing use-cases that demonstrate their application. Another goal is to modify the CTF competition exercises to be more attainable given the time allotted and will examine them to ensure a balance between categories in Mäses model for describing cybersecurity skills.

The rise of cybercrimes, the ongoing security breaches, the continuing threats of malware and ransomware, the growth of phishing and other online scams, and the ease in which misinformation can spread online all necessitate making students aware of cybersecurity issues, and teaching them to use OSINT tools to protect themselves and the organizations that will employ them, from cybersecurity attacks.

Teaching cybersecurity awareness in the university and training employees in the workplace can be a challenge due to the lack of experts in this field. Developing solutions, tools to automate the process, and activities that will spark students' interest will benefit students, teachers, and society at large. Ethics issues will emerge as users will need to trust systems using current technologies such as artificial intelligence, Internet of Things, or blockchain, that they may not fully understand. Universities also should look beyond their current cybersecurity needs to predict future developments and how to incorporate the impact of these and other current technologies in the cybersecurity awareness curriculum for information technology students.

6. REFERENCES

Ala-Mutka, K., Punie, Y., & Redecker, C. (2008). *Digital Competence for Lifelong Learning. Policy Brief*.

AP Computer Science Principles. (2017). Retrieved from https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf

Bada, M., Sasse, A. M., & Nurse, J. R. C. (2019). *Cyber Security Awareness Campaigns: Why do they fail to change behaviour?* 11.

Boulton, C. (2017, April 19). Humans are (still) the weakest cybersecurity link. Retrieved July 7, 2019, from CIO website: https://www.cio.com/article/3191088/humans-are-still-the-weakest-cybersecurity-link.html

Carretero-Gomez, S., Vuorikari, R., & Punie, Y. (2017, April 28). DigComp 2.1: The Digital Competence Framework for Citizens with eight proficiency levels and examples of use [Text]. Retrieved July 3, 2019, from EU Science Hub - European Commission website: https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/digcomp-21-digital-competence-framework-citizens-eight-proficiency-levels-and-examples-use

Chinien, C., & Boutin, F. (2011). *Defining Essential Digital Skills in the Canadian Workplace: Final Report*. 87.

Dawson, J., & Thomson, R. (2018). The Future Cybersecurity Workforce: Going Beyond Technical Skills for Successful Cyber Performance. *Frontiers in Psychology*, *9*. https://doi.org/10.3389/fpsyg.2018.00744

Dijk, J. van, & Deursen, A. van. (2014). *Digital skills: Unlocking the information society* (First edition). New York, NY: Palgrave Macmillan.

Frydenberg, M., & Press, L. (2010). From Computer Literacy to Web 2.0 Literacy: Teaching and Learning Information Technology Concepts Using Web 2.0 Tools. *Information Systems Education Journal*, *8*(10). Retrieved from https://eric.ed.gov/?id=EJ1146965

Glassman, M., & Kang, M. J. (2012). Intelligence in the internet age: The emergence and evolution of Open Source Intelligence (OSINT). *Computers in Human Behavior*, *28*(2), 673–682. https://doi.org/10.1016/j.chb.2011.11.014

Harris, M. A., & Patten, K. P. (2015). Using Bloom’s and Webb’s Taxonomies to Integrate Emerging Cybersecurity Topics into a Computing Curriculum</title>. *Journal of Information Systems Education*, *26*(3), 219.

Johnson, A. P. (2012). *A short guide to action research* (4th ed). Upper Saddle River, N.J: Pearson.

Kissiah, M., & eInvestigator.com. (2019, February 4). Open Source Intelligence Tools and Techniques for Investigations. Retrieved July 7, 2019, from Private Investigator and Investigation Resources website: https://www.einvestigator.com/open-source-intelligence-tools/

Lorenz, B. (2018). CyberSec Stories 1st Edition. Retrieved July 2, 2019, from https://sites.google.com/view/tty-csgame/

Marquardson, J. (2018). Infrastructure Tools for Efficient Cybersecurity Exercises. *Information Systems Education Journal*, *16*(6), 23.

Marquardson, J., & Gomillion, D. (2018). Cyber Security Curriculum Development: Protecting Students and Institutions While Providing Hands-On Experience. *Information Systems Education Journal*, *16*(5), 12.

Mäses, S., Randmann, L., Maennel, O., & Lorenz, B. (2018). Stenmap: Framework for Evaluating Cybersecurity-Related Skills Based on Computer Simulations. In P. Zaphiris & A. Ioannou (Eds.), *Learning and Collaboration Technologies. Learning and Teaching* (Vol. 10925, pp. 492–504). https://doi.org/10.1007/978-3-319-91152-6_38

National-Cyber-Strategy.pdf. (2018, September). Retrieved July 4, 2019, from National Cyber Strategy website: https://www.whitehouse.gov/wp-content/uploads/2018/09/National-Cyber-Strategy.pdf

Postimees. (2019, April 1). Editorial: The weakest link in cyber security is human. Retrieved from https://arvamus.postimees.ee/6559321/juhtkiri-kuberturvalisuse-norgim-luli-on-inimene

Prensky, M. R. (2012). *From Digital Natives to Digital Wisdom: Hopeful Essays for 21st Century Learning*. Corwin Press.

Rosenblum, G. (2014, May 15). SF Traffic Sign Hacked To Warn Drivers Of "Godzilla Attack." Retrieved July 2, 2019, from https://sanfrancisco.cbslocal.com/2014/05/15/prank-san-francisco-street-hack-godzilla-warning-sf-traffic-sign-hacked-to-read-godzilla-attack/

Ruoti, S., Andersen, J., Heidbrink, S., O'Neill, M., Vaziripour, E., Wu, J., … Seamons, K. (2016). "We'Re on the Same Page": A Usability Study of Secure Email Using Pairs of Novice Users. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 4298–4308. https://doi.org/10.1145/2858036.2858400

Serapiglia, A. (2016). The Case for Inclusion of Competitive Teams in Security Education. *Information Systems Education Journal*, *14*(5), 25.

van Laar, E., van Deursen, A. J. A. M., van Dijk, J. A. G. M., & de Haan, J. (2017). The relation between 21st-century skills and digital skills: A systematic literature review. *Computers in Human Behavior*, *72*, 577–588. https://doi.org/10.1016/j.chb.2017.03.010

Wells, D., Conflict, M., & Gibson, H. (2017). *OSINT FROM A UK PERSPECTIVE: CONSIDERATIONS FROM THE LAW ENFORCEMENT AND MILITARY DOMAINS*. 32.

Wheeler, S. (2010, November 2). Digital literacy 1: What digital literacies? Retrieved July 7, 2019, from Learning with 'e's website: http://www.steve-wheeler.co.uk/2010/11/what-digital-literacies.html

Zimmermann, V., & Renaud, K. (2019). Moving from a 'human-as-problem" to a 'human-as-solution" cybersecurity mindset. *International Journal of Human-Computer Studies*. https://doi.org/10.1016/j.ijhcs.2019.05.005

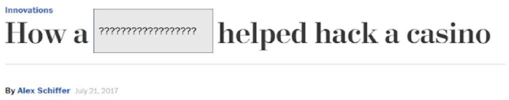
Appendix 1. OSINT Tools

Table 1. Open Source Intelligence Tools for Cybersecurity Awareness

Try this tool:	**To accomplish this task:**	**At this web address:**
Base64	Encode or decode data to / from base 64	https://www.base64encode.org/
BuiltWith	Determine a website's Content Management System and other technologies	https://builtwith.com
Check Usernames	Check availability of usernames on social networks	https://checkusernames.com/
Decode Ciphers	Encrypt / Decrypt SMS messages with T9 mode	https://www.dcode.fr/t9-cipher
Gaijin	Analyze Email Header to determine sender and recipient	https://www.gaijin.at/en/tools/e-mail-header-analyzer
Google Image Search Tin Eye	Reverse image search	https://images.google.com/ https://tineye.com
Have I Been Pwned?	Determine if your personal data has been compromised	https://haveibeenpwned.com/
IPLocation	Analyze IP address and details	https://www.iplocation.net/find-ip-address
Panopticlick	Determine if you are trackable in your browser	https://panopticlick.eff.org/
Phishing Quizzes	Learn about Phishing	https://phishingquiz.withgoogle.com/ https://www.sonicwall.com/en-us/phishing-iq-test-landing https://www.opendns.com/phishing-quiz/ https://accellis.com/phishing-quiz/
PhoneSpell	Encode a phone number to words	https://www.phonespell.org/
RandomUser UI Faces Fake Name Generator Fake Person Generator	Develop a fake identity online	https://randomuser.me/photos https://uifaces.co/ https://www.fakenamegenerator.com https://www.fakepersongenerator.com
Scam Advisor	Determine if a website is safe (http vs https)	https://www.scamadviser.com/
SleepingTime	Determine sleep patterns based on Twitter usage	http://sleepingtime.org/
Social Catfish	Find a person by a photo or social media information	https://socialcatfish.com/
VirusTotal	Analyze a suspicious file or web address to detect malware	https://www.virustotal.com
Web Mii Pipl	Find information about a person	http://webmii.com/ https://pipl.com/

Appendix 2. Sample CTF Exercises and Puzzles.

What word is missing?

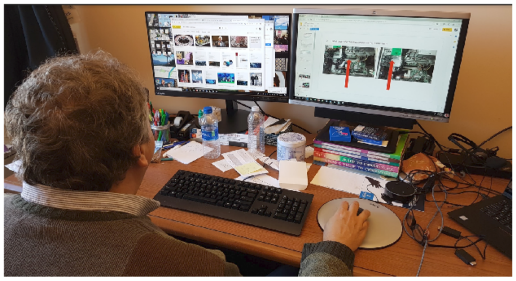


Innovations

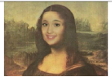
How a ?????????????????? helped hack a casino

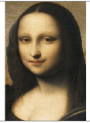
By Alex Schiffer July 21, 2017

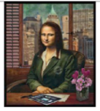
What's my password?

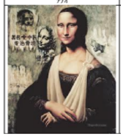


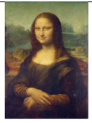
Create the sequence of numbers corresponding to photos that are not fakes.

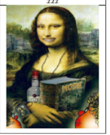
112


078


221


774


198


222


You have received a letter from your coach in the file list.exe

Hash is:
SHA-256:
24d004a104d4d54034dbcffc2a4b19a11f39008a575a
a614ea04703480b1022c

What does the message say?

We have intercepted a message from a well-known cybercriminal gang... try to decrypt it:

8444447777 4447777 66688777 2224426622233
933 633338 28 83366 76
7777337277733 99966688777777733555333

Crack a Wi-Fi Network

Open Wi-Fi networks can be a security risk. A service provider set up the Wi-Fi at your grandmother's house using an Asus RT-AC66U router.

The device is in the factory settings/configuration.

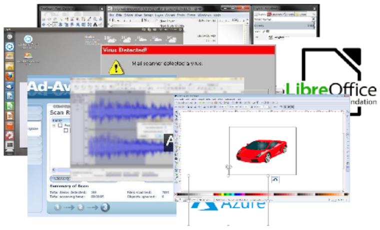
What is the default user name and password?

In a study with 10-12th graders, what is the correct descending order of situations that they have experienced?

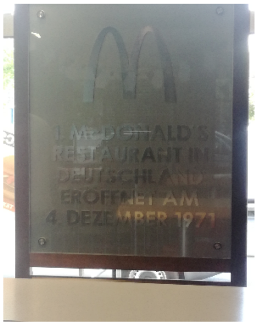
A. Friends shared pictures of me in public that I did not agree to
B. I posted publicly information that I should not
C. Never happened anything of that kind
D. My phone is missing or stolen
E. Someone accessed my data without permission
F. Someone hacked my social media account
G. Someone hacked my email

Know your programs
Pair the program topic and the original software name. Flag is the first letters of the programs original names. Be sure to use the original name or the program!!

- Office software
- Virus
- Operating System
- Vector graphic program
- Raster graphic program
- Audio program
- Antivirus
- Removes adware and spyware
- Cloud software



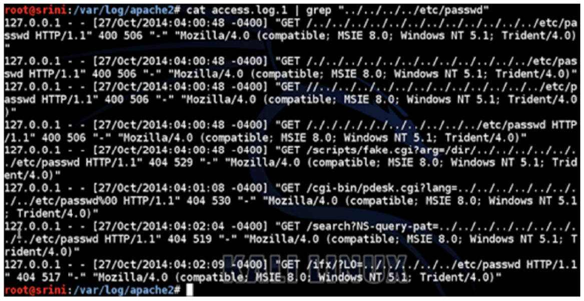
What's the address where this photo was taken?



- Analyze the domain name https://www.betterinternetforkids.eu/
- Who owns it?

- The LA Times website (www.latimes.com) is not visible for EU Citizens because of the GDPR challenges.
- List different ways how one can still see the website and find out news from example 1.09.2018 morning.

What happened?
What was accessed?
What is the name of this type of hack?



```
root@srini:/var/log/apache2# cat access.log.1 | grep "../../../../etc/passwd"
127.0.0.1 - - [27/Oct/2014:04:00:48 -0400] "GET /../../../../../../../../../../../etc/pa
sswd HTTP/1.1" 400 506 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)
"
127.0.0.1 - - [27/Oct/2014:04:00:48 -0400] "GET /./../../../../../../../../../../etc/pas
swd HTTP/1.1" 400 506 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)"
127.0.0.1 - - [27/Oct/2014:04:00:48 -0400] "GET //../../../../../../../../../../../etc/p
asswd HTTP/1.1" 400 506 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0
)"
127.0.0.1 - - [27/Oct/2014:04:00:48 -0400] "GET /./././././././../../../../etc/passwd HTTP
/1.1" 400 506 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)"
127.0.0.1 - - [27/Oct/2014:04:00:48 -0400] "GET /scripts/fake.cgi?arg=/dir/../../../../../.
./etc/passwd HTTP/1.1" 404 529 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trid
ent/4.0)"
127.0.0.1 - - [27/Oct/2014:04:01:08 -0400] "GET /cgi-bin/pdesk.cgi?lang=../../../../../../.
./../etc/passwd%00 HTTP/1.1" 404 530 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1
; Trident/4.0)"
127.0.0.1 - - [27/Oct/2014:04:02:04 -0400] "GET /search?NS-query-pat=../../../../../../../.
./../etc/passwd HTTP/1.1" 404 519 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; T
rident/4.0)"
127.0.0.1 - - [27/Oct/2014:04:02:09 -0400] "GET /ifx/?LO=../../../../etc/passwd HTTP/1.1
" 404 517 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)"
root@srini:/var/log/apache2# 
```

Academic Entitlement Beliefs of Information Systems Students: A Comparison with Other Business Majors and An Exploration of Key Demographic Variables and Outcomes

Scott J. Seipel
scott.seipel@mtsu.edu

Nita G. Brooks.
nita.brooks@mtsu.edu

Information Systems and Analytics
Middle Tennessee State University
Murfreesboro, TN

Abstract

Academic entitlement has received much attention in both academic and practitioner outlets. It is defined as "the tendency to possess an expectation of academic success without taking personal responsibility for achieving that success" (Chowning & Campbell, 2009 p. 982). The concept evolved from research in the area of generalized entitlement and narcissism resulting in a context-specific measure useful in understanding entitlement beliefs specific to educational environments. The overall goal of this research is to provide an introductory understanding of entitlement beliefs among information systems students and subsequently compare them to the greater population of students in a business college. Data was collected from 529 undergraduate students at a public university in the southeastern United States. A series of nested models were analyzed to better understand the overall structure of the construct and determine the extent of differences in the two populations. Additional demographic factors were examined including age, gender, employment status, and self-reported GPA (overall and within major). For the sample examined in the current study, findings indicated undergraduate information systems students are quite similar in their entitlement beliefs when compared to students in the other disciplines. Additionally, within-major GPA was found to be significantly related academic entitlement among both populations. A discussion of the findings is provided along with general recommendations for future research.

Keywords: academic entitlement; information systems students; student outcomes

1. INTRODUCTION

Over the past several years, there has been an increased focus on the view that the current generation of students feels more entitled to a college degree. This concept is referred to and operationalized as academic entitlement. It is defined as "the tendency to possess an expectation of academic success without taking personal responsibility for achieving that success" (Chowning & Campbell, 2009 p. 982).

Academic entitlement has been tied directly to a concept called consumerism. Sohr-Preston and Bosweel (2015) provided that in the context of higher education, consumerism represents a student's perspective that they are "paying customers for their education and deserve the same customer satisfaction and service as any other type of consumer" (p. 183). Essentially, this results in an exchange; the result of paying tuition is a degree and good GPA.

One of the driving goals in higher education environments remains the desire to understand students to more effectively promote and ensure learning and to guide them to successful completion of a degree. Understanding academic entitlement provides a means to help meet that goal.

The primary focus of this paper is to examine academic entitlement in undergraduate information systems students. Discipline specific studies are useful for many reasons. First, they help the discipline better understand its members, and second, they provide a frame of reference against which others can compare. Demographic factors are examined as well to determine where differences might exist. Specifically, factors included were gender, age, employment, major, and overall and within major GPA.

The following section presents a sample of literature that touches on the areas of generalized or psychological entitlement as well as academic entitlement with the primary focus being given to academic entitlement. The methodology, analysis, and results sections follow outlining the examination of academic entitlement for this sample. The paper then provides a discussion followed by directions for future research.

2. LITERATURE REVIEW

There have been several studies examining the notion of entitlement and closely related concepts such as the self-concept and self-esteem (Sohr-Preston and Boswell, 2015). Research focused on the organizational environment has highlighted entitlement as important due to the challenges it creates for managing today's workforce (Tomlinson, 2013).

Generalized or psychological entitlement has been studied in a variety of research domains. The concept of entitlement has been found in the literature as both a trait-like and state-like construct. Trait entitlement is defined as "a global sense of the privileges that is stable across time" (Tomlinson, 2013 p. 72). Specific contexts have also been examined in relation to entitlement. For example, research has been conducted examining entitlement related to the legal system, philosophy, political science, sociology, and other areas (Tomlinson, 2013). It represents the sense that individuals "ought to obtain a certain outcome" (Kopp, Zinn, Finney, & Jurich, 2011) or a general belief about what an individual deserves (Anderson, Halberstadt, & Aitken, 2013). Generalized entitlement is associated with narcissism and inflated views of the self-concept.

Specifically, the importance of entitlement and understanding the role it plays in general is highlighted by the negative behaviors associated with it in previous research. Campbell, et al. (2004) noted in their study aimed at developing a construct to measure psychological entitlement that entitlement has a "largely unconstructive impact on social behavior (p. 29). It has been found to be negatively related to factors such as agreeableness and stability (Jordan, Ramsay, & Westerlaken, 2017). When an outcome that is desired is not obtained by the individual, negative behaviors are likely when entitlement perceptions are higash (Kopp, et al., 2011). Additionally, entitlement has been found to be associated with positive behaviors such as making the choice to work for a "socially responsible organization" even though the choice would result in less pay (Thomason, Etling, Brownlee, & Charles, 2015).

The examination of entitlement expanded quite naturally to focus on the context of the academic arena. It is not uncommon to hear about the current generation of students being "entitled" and feeling that they deserve good grades or a degree – regardless of performance. Sohr-Preson and Boswell (2015) found that both academic dishonesty and external locus of control were significantly related to perceptions of academic entitlement. This ties to work conducted by Sessoms, et al. (2016) noting that students that are academically entitled exhibit certain "undesirable characteristics" (p. 1). These qualities include individual perceptions related to the amount of control the student has over the academic environment, an external locus of control, and the view, as noted earlier, that the student is a customer of the academic institution. As defined by Ajzen (2002), an external locus of control represents the perception that "outcomes are determined by nonbehavioral factors" (p. 676). This could essentially mean, that in the context of the academic environment, the outcomes (grades, etc.) are not perceived a result of specific behavior conducted by the student.

Expanding the examination of generalized or psychological entitlement to the academic environment has created much interest and has resulted in a context-specific construct aimed at understanding perceptions and beliefs of students in higher education. Several studies have looked at academic entitlement and have indicated its potential in explaining outcomes (e.g. Jordan, et al., 2017; Sessoms, et al., 2016). Using a measure specific academic entitlement, described in the following section, this study aims to provide additional detail related to how information system student performance and entitlement perceptions are related.

3. METHODOLOGY

Academic entitlement was assessed using the eight-item single-factor scale developed by Kopp et al. (2011). The items, shown in the Appendix in Table 1, were measured on a 7-point Likert-type scale with 1 representing "Strongly Disagree" and 7 representing "Strongly Agree".

The survey collected additional information including demographic data on gender, age, employment, major area of study, and year (academic classification) in school. Respondents also self-reported their overall GPA as well as their GPA in courses within their major area of study. GPA was collected in nine ordinal categories rather than as a raw value (Appendix Table 2).

Surveys were distributed to students at a large public university in the southeastern United States. The primary point of data collection was during an undergraduate course in business analytics that is required in programs for all majors in the college of business.

The voluntary survey was completed by 529 students which represent 24.7% of the population of students that would potentially be eligible to take that level of course. Of the submitted surveys, ten were removed from the sample due to lack of answers to items that were critical to the analysis, resulting in a final sample size of 519 students.

Table 2 (see Appendix) shows descriptive statistics on the demographic information collected in the survey as well as the proportion of certain characteristics in the population of students in the college of business. While the gender and major area appear to be fairly represented relative to the population, the academic classification and overall GPA differ substantially. As the course is a junior-level course, it would be expected that fewer sophomores would be eligible and that might skew the results towards upperclassmen and more specifically juniors. Concerning the self-reported overall GPA, the students appear to have systematically overestimated their academic performance despite the reporting of GPA in their semester grade report. It can be assumed that the same overestimation would occur with the self-reported GPA within their major area of study. It was noted that the overestimation of GPA was persistent even when the underrepresented sophomores were excluded from the population percentages.

4. ANALYSIS

In order to determine the degree of fit of the academic entitlement construct, an initial confirmatory factor analysis (CFA) was performed. The fit of the model was to be determined by the following combination of measures: 1) the χ^2 statistic; 2) the root mean square error of approximation (RMSEA; Steiger & Lind, 1980); 3) the comparative fit index (CFI; Bentler, 1990); and 4) the non-normed fit index (NNFI; Tucker & Lewis, 1973). Based on the advice of Hu and Bentler (1999), a value of .06 or below is considered an acceptable fit for the RMSEA, with comparative values of .90 or more (.95 or greater preferred) for the CFI and NNFI. All analyses were performed utilizing the lavaan package (Rosseel, 2012) and R (R Core Team, 2013).

The procedure used for this analysis began with a determination of overall fit of the CFA model. If a positive fit is achieved, the next step is to ascertain if group differences exist in the fit based on a student majoring in information systems relative to other majors. These differences can manifest themselves in multiple places in a CFA model, so a series of measurement models (Milfont and Fischer, 2010) are fit with increasing restrictions on the different components of the model that are allowed to vary among the groups. In general, six models are fit in sequence. Model 1 is the baseline model and incorporates the groups into the model with no restriction other than equivalent factorial structure. Configural invariance would be indicated if Model 1 shows good fit. Model 2, which includes the factor structure constraint from Model 1, adds the restriction of equivalent factor loadings among the groups. Metric invariance is achieved with a good Model 2 fit, and this would allow for the investigation of group differences in academic entitlement. Model 3 builds on Model 2 by adding a requirement for equal intercepts and is an indication of scalar invariance. Model 4 is a measure of strict model invariance by adding the restriction of equivalent error variances among the groups. Note group scores can be compared without the proper fit of Model 4. Models 5 and 6 are incremental to Model 4 and measure marginal change from that base. Model 5 tests the equivalence of factor variance/covariance structures among the groups. Model 6 evaluates the factor means to determine if they can be considered equal among the groups.

As the results of these models are incremental, the extent to which the academic entitlement factor differs among the groups can be determined by looking at the marginal changes in certain fit statistics. In other words, when the additional restriction in a subsequent model produces a reduced fit, then the preceding model provides an indication of the extent to which the groups do not vary. To evaluate these models, specific fit statistics designed for nested models are employed. In addition to those mentioned earlier, the Akaike Information Criterion (AIC; Akaike, 1974) and McDonald's non-centrality index (NCI; McDonald, 1989) fit statistics will be utilized. In general, higher values of AIC indicate a reduced fit. Cheung and Rensvold (2002) recommend that marginal changes .01 and .02 or more (on the negative scale) in the CFI and NCI measures respectively are indicative of reduced fit in the more restricted model.

Following the determination of any factor structure differences among the groups based on major area, an investigation was made to determine if demographic measures included in the study are associated with the academic entitlement level of the respondents. As gender and employment status are represented in groups, the procedure outlined above was utilized to determine if there are differences in academic entitlement structure among those factor levels. For quantitative variables age, overall GPA, and within-major GPA, composite academic entitlement scores were calculated for each respondent and regressed on those measures.

5. RESULTS

The internal consistency of the academic entitlement scale as measured by Cronbach's Alpha was .79. A maximum-likelihood confirmatory factor analysis was run on the sample for the first-order latent variable of academic entitlement. Overall model fit was acceptable, with $\chi^2 = 536.61$ (20 *df*, *p* = .000), RMSEA = .057 (90% CI = .039 -.076), CFI = .962, and NNFI = .946. All *p*-values of estimated parameters were at .000. The ratio of observations per estimated parameter was greater than 32 to 1, significantly more than the minimum of 5 to 1 suggested by Bentler and Chou (1987).

The academic entitlement CFA model was evaluated to determine if it was invariant to whether or not the student was majoring in information systems. Model fit statistics for the incremental Models 1 through 6 are shown in the Appendix in Table 3. Based on the results from Model 1, it can be concluded that the overall fit of the academic entitlement CFA model to students from the college was acceptable when their major in information systems (or not) is brought in as a mitigating factor. Results from subsequent Models 2 through 6 show that incremental restrictions were not significantly detrimental to the model's fit. All models show acceptable fit levels and marginal changes to AIC, CFI, and NCI are within acceptable values at all increments. Given these results, it can be concluded that the choice of the information systems major is not related to the level of academic entitlement in this population.

As the major areas have differing proportions of gender (e.g. males make up 75.7% of information systems majors yet 60% of all majors in this college), the academic entitlement CFA model was investigated to determine if it was invariant to gender. It was important to rule out that a difference in academic entitlement by major area was offset by a gender effect. As such, Models 1 through 6 were fit to the entitlement CFA model using gender as a mitigating factor. Model 1 showed acceptable fit (Appendix Table 4) with subsequent Models 2 through 6 showing no significant degradation in fit despite the additional constraints on invariance. It can be concluded that there is no significant difference in the academic entitlement model among genders and thus the invariance of the model to the information systems major was not gender related.

To investigate whether employment status was related to academic entitlement, an initial model was created that separated the three employment levels into groups to determine if there was a difference. The initial model showed some reduction in fit, particularly in the RMSEA fit statistic, which was beyond acceptable range at .069 (Appendix Table 5). Other fit statistics remained marginally acceptable, but subsequent Models 2 through 6 did not show marked change from the initial model as succeeding parts of the

CFA model were made invariant. From these results, it was concluded that the employment status of a student was unrelated to their level of academic entitlement.

To determine if academic entitlement was related to the other demographic factors, composite scores for academic entitlement were calculated using the coefficient matrix from the base confirmatory factor analysis on academic entitlement (Appendix Table 1). The composite scores were regressed on the age of the student and the self-reported GPAs. As the overall and within-major GPAs were recorded using an ordinal scale, the midpoint of each GPA category was utilized to create an approximate estimate. The fit of this model indicated a significant inverse relationship between the mean academic entitlement score and age ($p \approx .0101$) and within-major GPA ($p < .001$). Interestingly, the relationship with overall GPA was not found to be significant ($p \approx .0995$) nor inverse. However, subsequent investigation of the within-major GPA showed that overall GPA became significant ($p \approx .0415$) with an inverse relationship when within-major GPA was removed from the model; they simply shared information as would be expected. R-squared for the initial regression model was .0569.

As a final comparison of students majoring in information systems with those that are not, a model that included academic entitlement with age, within-major GPA, and gender was created. The initial fit of this model was acceptable (Appendix Table 6). The coefficient estimates for the covariates in this initial model did show some apparent difference as the information systems students had a significant inverse relationship between academic entitlement and age ($p \approx .005$), and academic entitlement and within-major GPA ($p \approx .019$). Students in other majors had a significant inverse relationship with academic entitlement and within-major GPA ($p < .001$), but the relationship with age was insignificant ($p \approx .152$). In both major areas, the relationship of gender to academic entitlement was not significant ($p > .500$). To test the equivalence of the significant relationships, a seventh model was added to Models 1 through 6 to specifically test the invariance of regression slopes among the two groups. Results from the series of models seemed to show no apparent difference in the groups even among the regression slopes. In conclusion, there was insufficient evidence to show that students majoring in information systems are different from other majors in academic entitlement and its relationship to age, within-major GPA, and gender.

6. DISCUSSION

The importance and potential power of academic entitlement has been noted. Studies have examined both generalized and psychological entitlement for many years. The inclusion of a specific measure to be used in academic environments highlights its importance as well as the need to take a context specific approach.

As the overall goal was to examine entitlement for information systems students, the study allowed for the inclusion of additional majors that made the exploration more successful. Being able to compare across groups has driven numerous studies in the IS discipline. While the findings indicated that the groups were similar, this does help universities and those in education form a general perspective. Just because the groups are similar does not take away from the potential of academic entitlement to impact outcomes.

It was interesting to find that age was not found to be significantly related to academic entitlement. This would indicate that at least among current students, generational differences are not apparent, which seems counter to what is perceived. Entitlement is more connected to performance, or, more specifically, the lack of performance academically. Perhaps poorer students see the scores of higher performance students, desire them, and consequentially feel entitled to them too. Previous research has shown that individuals that underperform often have higher levels of academic entitlement (Anderson, et al., 2013). Higher academic performance students may feel they earned their scores through effort.

Previous research had found gender differences in academic entitlement (Ciani, Summers, & Easter, 2008; Sohr-Preston & Boswell, 2015), but this study did not replicate those findings. Gender did not play a role in either the information systems major or the group of students in other majors in the business college. In comparing the two studies, it is important to note that different measurement instruments were used (Achacoso, 2002; Chowning & Campbell, 2009), and it was not possible to compare other demographic factors across the groups.

Our findings highlighted the role of within-major GPA as being related to the measure of academic entitlement used in this study. While this may seem like a minor finding, it could point to the potential impact for academic entitlement beliefs to be stronger towards the major when compared to situations that are not major specific.

7. LIMITATIONS AND FUTURE RESEARCH

As with any study, there is a need to address limitations and options for improving future research in the area. The cross-sectional nature of the data used warrants attention. The data for this study was collected at a single point in a course geared to the junior-level of a student's academic program. It would be necessary, to fully understand the importance of academic entitlement, to collect data at multiple points in time. This would allow for additional exploration related to the relationship between entitlement and performance. Academic entitlement, as previously noted, is a contextual construct rooted in concepts provided by personality studies (narcissism) and other factors related to the self-concept. While often stable, trait-like constructs can and do change over time. Since academic entitlement is specific to the academic environment, it is possible that perceptions change as an individual progress through the chosen course(s) of study. In a study conducted by Sessoms, et al. (2016), findings indicated there could be increases over time, but the authors noted additional research should be conducted.

It would be beneficial to collect data from multiple higher education institutions. This study focused on data collected from one institution. Collecting data from students at other public as well as private universities would strengthen understanding of the construct and the role it plays in student behaviors and outcomes.

The GPA used in this study, as a measure of student performance, was reported by the respondent. This could be a potential issue and may be addressed by collecting the data directly from the institutions. It is also necessary to expand the examination of academic entitlement to include other outcomes as well as factors that influence these perceptions. Understanding the relationship to satisfaction or other outcomes for information systems majors with the academic experience would be interesting. As noted earlier, students often view the university as a consumer would when purchasing a product at a retail store. Academic entitlement would seemingly play a role in the evaluation of the program attended just as it has been noted to play a role in the general evaluation.

8. CONCLUSION

The goals of this study were to gain a better understanding of academic entitlement in undergraduate information systems students, to determine whether academic entitlement differed across key demographic variables, and to examine whether there was a relationship with outcomes (GPA). The sample allowed for additional analysis of undergraduate students in other business disciplines as well as a comparison of IS students to other majors. After completing analyses on several models, results indicated that academic entitlement was related to within-major GPA for the students examined. While there were no additional significant differences between majors in this study or across the demographic factors included, the importance of understanding academic entitlement in higher education remains. The focus on the IS student allowed for a comparison, which is often seen as necessary. Historically, individuals in the IS profession have been viewed as unique; therefore, we tend to carry that concept forward making sure to always validate similarities or highlight inconsistencies. In this case, the primary path to follow is to include academic entitlement beliefs in situations where you are trying to assess performance (real and perceived) and in situations where any outcomes play a role. If concepts introduced about the construct (changing over time, correlating with negative behaviors, etc.) prove to be consistent, there could be significant change warranted in higher education.

8. REFERENCES

Achacoso, M.V. (2002). What do you mean my grade is not an A? An investigation of academic entitlement, causal attributions, and self-regulation in college students. Unpublished Doctoral Dissertation. University of Texas, Austin.

Ajzen, I. (2002). Perceived behavioral control, self-efficacy, locus of control, and the Theory of Planned Behavior. *Journal of Applied Social Psychology*, 32(4), 665-683.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716–723.

Anderson, D., Halberstadt, J., & Aitken, R. (2013). Entitlement attitudes predict students' poor performance in challenging academic conditions. *International Journal of Higher Education*, 2(2), 151-158.

Bentler, P.M. (1990). Comparative Fit Indexes in Structural Models. *Psychological Bulletin*, 107, 238-246.

Bentler, P.M. & Chou, C.P. (1987). Practical issues in structural modeling. *Sociological Methods & Research*, 16, 78-117.

Campbell, W.K., Bonacci, A.M., Shelton, J., Exline, J.J., & Bushman, B.J. (2004). Psychological entitlement: interpersonal consequences and validation of a self-report measure. *Journal of Personality Assessment*, 83(1), 29-45.

Chowning, K. & Campbell, N.J. (2009). Development and validation of a measure of academic entitlement: individual differences in students' externalized responsibility and entitled expectations. *Journal of Educational Psychology*, 101(4), 982-997.

Cheung, G.W. & Rensvold, R.B. (2002). Evaluating Goodness-of-Fit Indexes for Testing Measurement Invariance, *Structural Equation Modeling: A Multidisciplinary Journal*, 9(2), 233-255.

Ciani, K.D., Summers, J.J., & Easter, M.A. (2008). Gender differences in academic entitlement among college students. *The Journal of Genetic Psychology*, 169(4), 332-344.

Duckworth, A.L., Peterson, C., Matthews, M.D., Kelly, D.R. (2007). Grit: Perseverance and passion for long-term goals. *Journal of Personality and Social Psychology*, 92(6), 1087-1101.

Finney, T.G. & Finney, R.Z. (2010). Are students their universities' customers? An exploratory study. *Education + Training*, 52(4), 276-291.

Hu, L. & Bentler, P.M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling*, 6(1), 1-55.

Jeffres, M.N., Barclay, S.M., Stolte, S. (2014). Academic entitlement and academic performance in graduating pharmacy students. *American Journal of Pharmaceutical Education*, 78(6), 1-9.

Jordan, P.J., Ramsay, S., & Westerlaken, K.M. (2017). A review of entitlement: implications for workplace research. *Organizational Psychology Review*, 7(2), 122-142.

Kopp, J.P., Zinn, T.E., Finney, S.J., & Jurich, D.P. (2011). The Development and Evaluation of the Academic Entitlement Questionnaire. *Measurement and Evaluation in Counseling and Development*, 44(2), 105-129.

McDonald, R.P. (1989). An index of goodness-of-fit based on non-centrality. *Journal of Classification*, 6, 97-103.

Milfont, T.L. & Fischer, R. (2010). Testing measurement invariance across groups: Applications in cross-cultural research. *International Journal of Psychological Research*, 3(1), 111-121.

R Core Team (2013). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, 48(2), 1-36. URL http://www.jstatsoft.org/v48/i02/.

Sessoms, J., Finney, S.J., & Kopp, J.P. (2016). Does the measurement of academic entitlement change over time? *Measurement and Evaluation in Counseling and Development*, 49(3), 243-257.

Sohr-Preston, S. & Boswell, S.S. (2015). Predicting academic entitlement in undergraduates. *International Journal of Teaching and Learning in Higher Education*, 27(2), 183-193.

Steiger, J.H. & Lind, J.C. (1980, May). *Statistically-based tests for the number of common factors*. Paper presented at the Annual Spring Meeting of the Psychometric Society, Iowa City, IA.

Thomason, S., Etling, C., Brownlee, A., & Charles, T. (2015). Psychological entitlement and conscientiousness as predictors of socially responsible workplace decisions. *Small Business Institute Journal*, 11(2), 1-15.

Tomlinson, E.C. (2013). An integrative model of entitlement beliefs. *Employee Responsibilities and Rights Journal*, 25, 67-87.

Tucker, L.R. & Lewis, C. (1973). A reliability coefficient for maximum likelihood factor analysis. *Psychometrika*, 38, 1-10.

9. Appendix

Table 1: Academic Entitlement Items

Item	Statement	Mean	Std. Dev.	CFA Coefficients
1	If I don't do well on a test, the professor should make tests easier or curve grades.	3.68	1.68	1.000
2	If I am struggling in a class, the professor should approach me and offer to help.	3.57	1.84	0.919
3	If I cannot learn the material for a class from lecture alone, then it is the professor's fault when I fail the test.	2.61	1.53	0.731
4	I am a product of my environment. Therefore, if I do poorly in class, it is not my fault.	2.25	1.40	0.756
5	Because I pay tuition, I deserve passing grades.	2.05	1.48	0.853
6	Professors should only lecture on material covered in the textbook and assigned readings.	3.21	1.78	0.867
7	It is the professor's responsibility to make it easy for me to succeed.	2.77	1.64	1.073
8	I should be given the opportunity to make up a test, regardless of the reason for the absence.	3.34	1.85	1.063

Table 2: Demographics

Variable	Sample	Sample Percent	Percent in College
Year			
Sophomore	14	2.7%	24.5%
Junior	301	58.4%	33.6%
Senior	200	38.8%	42.0%
Missing	14		
Gender			
Male	317	60.0%	58.8%
Female	211	40.0%	41.2%
Missing	1		
Major Area			
Accounting	83	15.7%	18.8%
Business Administration	114	21.6%	23.4%
Economics	7	1.3%	2.6%
Entrepreneurship	16	3.0%	3.1%
Finance	59	11.2%	11.7%
Information Systems	100	18.9%	16.1%
Marketing	54	10.2%	12.1%
Management	82	15.5%	9.6%
Other	14	2.6%	2.6%
GPA Overall			
Below 2.00	0	0.0%	3.2%
2.00-2.24	23	4.4%	5.9%
2.25-2.49	19	3.6%	11.2%
2.50-2.74	44	8.4%	14.5%

2.75-2.99	72	13.8%	15.6%
3.00-3.24	123	23.5%	16.1%
3.25-3.49	86	16.4%	14.4%
3.50-3.74	73	14.0%	10.7%
3.75-4.00	83	15.9%	8.5%
Missing	6		
GPA within Major			
Below 2.00	3	0.6%	
2.00-2.24	0	0.0%	
2.25-2.49	7	1.4%	
2.50-2.74	31	6.2%	
2.75-2.99	64	12.8%	
3.00-3.24	125	25.0%	
3.25-3.49	79	15.8%	
3.50-3.74	84	16.8%	
3.75-4.00	107	21.4%	
Missing	29		
Employment			
Full-time	128	25.7%	
Part-time	270	54.2%	
Not Employed	100	20.1%	
Missing	31		
Age			
19-21	103	47.9%	
22-24	72	33.5%	
25-27	21	9.8%	
28-30	8	3.7%	
31-33	5	2.3%	
34-36	4	1.9%	
37 or more	2	0.9%	
Missing	14		

Table 3: Goodness of Fit Statistics for Model Testing

Measurement Invariance across Major Area (INFS/Non-INFS)

Model	χ^2	*df*	p-value	RMSEA	AIC	CFI	NCI	NNFI
1	72.53	40	0.001	0.056	15087.4	0.963	0.969	0.948
2	80.09	47	0.002	0.052	15087.0	0.962	0.969	0.955
3	89.38	54	0.002	0.050	15076.3	0.960	0.966	0.958
4	103.51	62	0.001	0.051	15074.4	0.953	0.961	0.957
5	104.17	63	0.001	0.050	15073.1	0.953	0.961	0.958
6	103.76	63	0.001	0.050	15072.7	0.958	0.961	0.959

Comparison of Nested Models

Models	$\Delta\chi^2$	Δdf	p-value	Δ RMSEA	Δ AIC	Δ CFI	Δ NCI	Δ NNFI
1 to 2	7.56	7	0.373	-0.004	-0.4	-0.001	-0.001	0.007
2 to 3	9.29	7	0.232	-0.002	-10.7	-0.002	-0.002	0.003
3 to 4	14.13	8	0.078	0.001	-1.9	-0.007	-0.006	-0.001
4 to 5	0.66	1	0.417	-0.001	-1.3	0.000	0.000	0.001
4 to 6	0.24	1	0.623	-0.001	-1.8	0.005	0.001	0.002

Note: RMSEA = root mean square error of approximation; AIC = Akaike Information Criterion; CFI = comparative fit index; NCI = McDonald's non-centrality index; NNFI = non-normed fit index. Model 1 = equality of overall structure; Model 2 = Model 1 plus invariant loadings; Model 3 = Model 2 plus equivalent intercepts; Model 4 = Model 3 plus invariant residuals; Model 5 = Model 4 plus invariant factor covariance matrices; Model 6 = Model 4 plus invariant factor means.

Table 4: Goodness of Fit Statistics for Model Testing

Measurement Invariance across Gender

Model	χ^2	*df*	p-value	RMSEA	AIC	CFI	NCI	NNFI
1	73.66	40	0.001	0.057	15051.2	0.962	0.968	0.947
2	89.64	47	0.000	0.059	15053.1	0.952	0.960	0.943
3	107.07	54	0.000	0.062	15056.6	0.940	0.950	0.938
4	112.86	62	0.000	0.056	15046.4	0.943	0.952	0.948
5	113.68	63	0.050	0.056	15045.2	0.943	0.952	0.949
6	112.95	63	0.000	0.055	15044.4	0.944	0.953	0.950

Comparison of Nested Models

Models	$\Delta\chi^2$	Δdf	p-value	Δ RMSEA	Δ AIC	Δ CFI	Δ NCI	Δ NNFI
1 to 2	15.98	7	0.025	0.002	2.0	-0.010	-0.008	-0.004
2 to 3	17.43	7	0.015	0.003	3.4	-0.012	-0.010	-0.005
3 to 4	5.80	8	0.670	-0.006	-10.2	0.003	0.002	0.010
4 to 5	0.81	1	0.368	0.000	-1.2	0.000	0.000	0.001
4 to 6	0.09	1	0.763	-0.001	-1.9	0.001	0.001	0.002

Note: RMSEA = root mean square error of approximation; AIC = Akaike Information Criterion; CFI = comparative fit index; NCI = McDonald's non-centrality index; NNFI = non-normed fit index. Model 1 = equality of overall structure; Model 2 = Model 1 plus invariant loadings; Model 3 = Model 2 plus equivalent intercepts; Model 4 = Model 3 plus invariant residuals; Model 5 = Model 4 plus invariant factor covariance matrices; Model 6 = Model 4 plus invariant factor means.

Table 5: Goodness of Fit Statistics for Models Testing

Measurement Invariance across Gender

Model	χ^2	*df*	p-value	RMSEA	AIC	CFI	NCI	NNFI
1	73.66	40	0.001	0.057	15051.2	0.962	0.968	0.947
2	89.64	47	0.000	0.059	15053.1	0.952	0.960	0.943
3	107.07	54	0.000	0.062	15056.6	0.940	0.950	0.938
4	112.86	62	0.000	0.056	15046.4	0.943	0.952	0.948
5	113.68	63	0.050	0.056	15045.2	0.943	0.952	0.949
6	112.95	63	0.000	0.055	15044.4	0.944	0.953	0.950

Comparison of Nested Models

Models	$\Delta\chi^2$	Δ*df*	p-value	Δ RMSEA	Δ AIC	Δ CFI	Δ NCI	Δ NNFI
1 to 2	15.98	7	0.025	0.002	2.0	-0.010	-0.008	-0.004
2 to 3	17.43	7	0.015	0.003	3.4	-0.012	-0.010	-0.005
3 to 4	5.80	8	0.670	-0.006	-10.2	0.003	0.002	0.010
4 to 5	0.81	1	0.368	0.000	-1.2	0.000	0.000	0.001
4 to 6	0.09	1	0.763	-0.001	-1.9	0.001	0.001	0.002

Note: RMSEA = root mean square error of approximation; AIC = Akaike Information Criterion; CFI = comparative fit index; NCI = McDonald's non-centrality index; NNFI = non-normed fit index. Model 1 = equality of overall structure; Model 2 = Model 1 plus invariant loadings; Model 3 = Model 2 plus equivalent intercepts; Model 4 = Model 3 plus invariant residuals; Model 5 = Model 4 plus invariant factor covariance matrices; Model 6 = Model 4 plus invariant factor means.

Table 6: Goodness of Fit Statistics for Models Testing

Measurement Invariance across Major Area (INFS/Non-INFS) including Major GPA, Age, and Gender

Model	χ^2	*df*	p-value	RMSEA	AIC	CFI	NCI	NNFI
1	147.13	82	0.000	0.057	14433.6	0.927	0.936	0.907
2	154.57	89	0.000	0.055	14427.1	0.926	0.936	0.914
3	164.38	96	0.000	0.054	14422.9	0.923	0.933	0.917
4	176.34	104	0.000	0.053	14418.8	0.919	0.929	0.919
5	176.36	105	0.000	0.052	14416.9	0.920	0.930	0.921
6	177.71	105	0.000	0.053	14418.2	0.918	0.929	0.919
7	179.49	107	0.000	0.052	14416.0	0.919	0.929	0.921

Comparison of Nested Models

Models	$\Delta\chi^2$	Δ*df*	p-value	Δ RMSEA	Δ AIC	Δ CFI	Δ NCI	Δ NNFI
1 to 2	7.44	7	0.384	-0.002	-6.5	-0.001	0.000	0.007
2 to 3	9.81	7	0.200	-0.001	-4.2	-0.003	-0.003	0.003
3 to 4	11.96	8	0.153	-0.001	-4.1	-0.004	-0.004	0.002
4 to 5	0.02	1	0.893	-0.001	-1.9	0.001	0.001	0.002
4 to 6	1.37	1	0.242	0.000	-0.6	-0.001	0.000	0.000
4 to 7	3.15	3	0.369	-0.001	-2.8	0.000	0.000	0.002

Note: RMSEA = root mean square error of approximation; AIC = Akaike Information Criterion; CFI = comparative fit index; NCI = McDonald's non-centrality index; NNFI = non-normed fit index. Model 1 = equality of overall structure; Model 2 = Model 1 plus invariant loadings; Model 3 = Model 2 plus equivalent intercepts; Model 4 = Model 3 plus invariant residuals; Model 5 = Model 4 plus invariant factor covariance matrices; Model 6 = Model 4 plus invariant factor means; Model 7 = Model 4 plus invariant regression slopes.

An Assignment a Day Scaffolded Learning Approach for Teaching Introductory Computer Programming

Deepak Dawar
daward@miamioh.edu

Marianne Murphy
murph103@miamioh.edu

Miami University
Hamilton, Ohio

Abstract

Teaching introductory programming courses to university students who come from a varied set of academic and non-academic backgrounds is challenging. Students who are learning programming for the first time can become easily discouraged leading to procrastination that subsequently can have an unfavorable effect on their learning outcomes, and overall final grade. This work proposes An Assignment A Day (AAAD) Scaffolded Learning approach, and presents our experiences with this pedagogical approach. According to neuroscience research, when subjects are engaged continuously with a task, there is improvement in the brain's neuroplasticity. Based on this research and our own experiences with entry level programming students, we pursued the research question: "Can a targeted continuous engagement with course material, and problem solving assignments improve learning outcomes?" The students, instead of writing an assignment and a lab for each module, were asked to complete one assignment a day, not exceeding four assignments a week. The limited areas of impact that we targeted were student procrastination in submitting assignments, student failure to submit assignments, and student engagement. The overall acceptance of this technique by students has been quite positive, and we report an improvement in assignment submission rates, and final exam scores, apart from improved student engagement. Students found the approach extremely effective in spite of having to spend considerable amount of time on assignments almost everyday.

Keywords: Introductory level programming, pedagogy, student engagement, neuroplasticity, student procrastination, learned helplessness.

1. INTRODUCTION

Introductory programming is an arduous process for many students especially those who have little or no prior experience. Low course completion rates are consistently reported (Bennedsen & Caspersen, 2007; Newman, Gatward, & Poppleton, 1970; Allan & Kolesar, 1997; (Sheard, & Hagan, 1998; Beaubouef & Mason, 2005; Kinnunen & Malmi 2006; Howles, 2009; Mendes et al., 2012; Watson & Li, 2014). Apart from learning and recognizing the syntax and semantics of the programming language, one also has to create a mental model of the solution (Sorva, 2013). The novice programmer has to grapple with multiple domains of learning as suggested in the literature (Rogalski & Samurçay, 1990; Kim & Lerch, 1997; Robins, Rountree, & Rountree, 2003; Davies, 1993).

It has also been suggested that the most difficult aspect faced by novice programmers may not be related to the specifics of the language at all. According to Lahtinen, Ala-Mutka, & Järvinen, 2005, understanding how to design a program, and dividing functionality into procedures are the primary problems faced by entry level programming students. Further, even after successful course completion, student learning in these introductory programming courses is not always retained (McCracken et al., 2001; Utting et al., 2013). Does that mean that programming as a course is more difficult than other similar level courses? There is no consensus on this theory, but there is a large body of data to suggest that this might be the case (Luxton-Reilly, 2016). In-fact, when computing courses were studied under the framework of two prominent taxonomies i.e. SOLO (Brabrand & Dahl, 2009), and BLOOM (Oliver et al., 2004) these courses were found to be more challenging than other courses. A recent study by Margulieux, Catrambone & Schaeff er.,2018, compared the domain difficulty of three courses – computer programming, chemistry, and statistics, and found computer programming to be the most difficult of three due to the complexity of the content to be learned and handled at a given time.

The authors of this paper have faced similar challenges in their classrooms while teaching introductory programming classes. From less than desirable passing rates, to inability of students to apply the learned concepts in subsequent programming classes led us to investigate the reasons more closely as relevant to our classroom setup, and provide possible interventions and remedies. This work is the result of one such intervention. The authors observed that one of the primary reasons for learning outcome failures in the class was student’s procrastination and lack of motivation to finish the assignment(s) on time. Motivation is a vast subject in its own right, and can take myriad forms.

We suspect that the lack of motivation and procrastination may just be symptoms of an abnormal cognitive load that programming assignments, and related tasks carry for many students. Cognitive load theory (Sweller, 1988, 1994; Paas, Renal, & Sweller, 2003; Plass, Moreno, & Brünken, 2010) deals with the aspects of load placed on working memory while a task is being executed. The amount and nature of this load depends upon the interactive nature of elements involved in the tasks. Computer programming requires balancing numerous interactive tasks. For example, writing a computer program involves juggling numerous details like problem domain, current state of program, language syntax, strategies etc. (Winslow, 1996).

The landscape of the potential problems faced by novice programmers is vast, and is quite formidable. Instead of dealing with the motivational aspect of programming directly, we turned to an approach that couples program scaffolding with the generally accepted notion that constant practice improves the learning outcomes, and as shown by psychological studies (Brown & Bennett, 2002; Moors & De Houwer, 2006; Glover, Ronning, & Bruning, 1990) done on variable student populations. Constant practice can also make students want to learn more (Kalchman, Moss, & Case, 2001). Constant practice and improved problem solving skills have shown to be mutually dependent and shown to be in a complex relationship as shown by Eckerdal, 2009. There is a plethora of studies confirming the important role practice and experience play in developing problem solving strategies by novice programmers. In a series of studies conducted by Rist (1986, 1989, 1995, 2004), and reviewed by Sorva (2012) confirm that one of the main differentiator of students into novice and expert programmers is their constant engagement and experience with learned schemata.

Keeping these factors in mind, we designed An Assignment A Day (AAAD) Scaffolded Learning approach wherein students were given a programming assignment a day, and no more than four assignments a week. Every assignment built on the previous assignment(s), and the final assignment was to be a mini-project testing students on all the concepts learned so far in previous assignments. We are faced with a few dilemmas though. First, it has been shown that constant testing of students leads to high levels of anxiety that may lead to sub-optimal performance (Kaplan et. al, 2005). Second, solving hard problems can easily bring down the morale of the novice programmers, and may send them into the spiral of learned helplessness, leading to poor performance (Crego et. al, 2016). To mitigate these effects, and at the same time make the students practice as much as possible, we made sure that the opening assignment tests very basic concepts, and then subsequent assignments gradually increase in complexity. We opined that having assignments designed in increasing order of complexity will reduce cognitive load on students thereby possibly resulting in better learning outcomes.

This opinion was based, in part, on classroom observations, and a study conducted by Alexandron et al. (2014). This study demonstrated the effectiveness of aligning tasks in increasing order of complexity on cognitive load, though the mandate of the study was much wider than studying this correlation.

2. METHODOLOGY

We created An Assignment A Day (AAAD) Scaffolded programming approach for introductory programming courses for our student population. The main driver of this intervention was the observation that in the orthodox model (one assignment a module that we followed), many students tend to procrastinate, and delay working on the assignments as late as possible. When the submission deadline approaches, they jump into action. It is evidenced from our experience that quite a high number of questions from students are received in last three hours prior to submission deadline. They are then faced with multiple complexities of the assignment leading to increased cognitive load. This increased load may give rise to student frustration, unwillingness to continue to work on the assignment, and eventually may lead to unfavorable learning outcomes. The purpose of this intervention was to make students constantly practice the material thereby potentially improving their chances of learning the material. We opined that this approach will assert a slight positive stress on students to submit the assignment at the end of the day. We also realized that the possible success of this scheme will significantly depend upon rendering the cognitive load asserted by the assignments, germane or manageable. AAAD was designed keeping all these possibilities in mind.

Our method is quite simple – make the students practice constantly and assert just the optimum stress on them in terms of deadlines and materials, so as to avoid student disenchantment and frustration with the course, while simultaneously improving learning gains. Being run for the first time, and due to small sample size, we are not in a position to define as to what constitutes the optimal load, as of now.

We tried to keep the AAAD approach as straightforward as possible with a few exceptions in between. The approach can be summarized as:

1. Students will ideally do one assignment per day
2. Opening assignments of the chapter will test students on very basic skills like writing a method stub. Subsequent assignments will gradually increase in complexity keeping in mind the cognitive load asserted by the assignment
3. There will not be more than four assignments per week
4. The final assignment should test students on all the previously learned chapter concepts
5. As an exception, and depending upon the cognitive load, an assignment may be completed in two or more days rather than a single day. This should mainly apply to the last assignment of the chapter that tests students on multiple concepts, but can also extend to assignments that variably test single but difficult concepts, and are not the last assignment of the chapter.
6. All other factors like quizzes, projects etc. remain the same for experimental and the control group.

The study was conducted over two semesters. The control group data was collected in the first semester. This group worked with the orthodox approach followed at our institution for introductory programming classes i.e. on an average, one assignment and one lab per week, with quizzes at the end of the module/chapter.

In the next semester, the experimental group was administered the AAAD approach, and data collected at the end of semester. A total of 37 assignments were given to the experimental group over a course of 13 weeks of which 1 week was spring break. Rest of the 12 weeks meant 84 days of which weekends accounted for 24 days. 10 days were meant for quizzes and exams. Hence, the students had to complete 37 assignments in about 50 days i.e. about 0.75 assignments a day. An additional end of course survey was conducted with the experimental group to measure how well this approach was received by the students.

Student Population

The student population of our department consists of both traditional and non-traditional students, though the terms are not well defined in literature. For the purposes of this work, we define traditional as students who are full time, and are recent high school graduates. Non-traditional students are those who have full-time jobs, are part-time students, and/or are older, and seeking a new career for a variety of reasons.

The number of students in the control and the experimental group were 20 and 22 respectively. One student from the control group declined to have their data included in the study. The course is mandatory for Computer Science (CSE) students but can be used as an elective for Information Technology (IT) majors. The control group had 12 IT/CSE majors and 8 non IT/CSE students. The experimental group had 13 IT/CSE, and 9 non IT/CSE majors. So the class composition of both groups is fairly similar, with the control group and experimental group having about 40% and 41% non IT/CSE majors respectively. This relatively similar class composition gives us some confidence about the experimental set up. It could have been quite difficult to compare results, had the IT/CSE and non IT/CSE major ratios varied widely.

Sample Load

To describe the procedure effectively, a sample load is presented here. The chapter/module to be presented is "method writing" in JAVA. This was to be delivered as an eight-day module with classroom practice labs (non-graded), five assignments, and a quiz at the end. Here is brief a description of assignments. Detailed descriptions of these assignments are included in Appendix B. As can be seen from Table 1 (see Appendix A), even a slight modification of problem statement can quickly increase the number of concepts that the student has to deal with, thereby increasing the cognitive load. This issue, in our opinion has to be dealt with effectively, if we are to improve upon the chances of student learning.

Comparison

Since the experimental group had to do many more assignments (at least 4 more assignments per module), an equitable comparison between the groups was a challenge. We decided that the comparison of the last summative assignment given to the experimental group with the usual assignment given to the control group would make a fair comparison. Both these assignments were similar in terms of concepts they tested but there were also some differences. For example, they differed in cognitive load and total points in many cases. The experimental group students have had more exposure to the concepts since they would have submitted a series of assignments by now. Our intervention assessed the following metrics for both groups, and for each assignment compared.

- Late submissions
- No submissions

To measure the impact of our technique on overall grades, if any, we administered the exact same module quizzes, and final exam to both groups and compared the following data points for both groups:

- Module wise quiz scores
- Final exam scores

Apart from this inter group comparison; we also performed an intra-group comparison for the experimental group to track student performance within the module, and the course as a whole. Observations and results are listed, and analyzed in next section.

3. RESULTS

We divided our analyses into two parts - inter and intra group. Inter group analyses compared the control with the experimental group, and intra group analyzed just the experimental group.

Inter Group Analyses

The control group did only one assignment per week whereas the experimental group did several leading up to the last assignment of the module. We compared the statistics of the last module assignment with the usual assignment of the control group. As an example, for assignments listed in Table 1, in the control group, an assignment similar to 5 was given to the students. In the experimental group, however, the same assignment 5 was given as the last assignment, after students have had some exposure to the relevant concepts in the previous assignments vis-à-vis assignments 1, 2, 3, and 4.

Table 2, 3 and 4 summarize the data points collected for comparison. The number of possible submissions per module in the control and experimental groups were 20 and 22 respectively.

No Submissions		
Module	Control (20)	Experimental (22)
1	1	0
2	0	0
3	0	0
4	2	0
5	2	1
6	5	3
7	4	3
Total	14	7

Table 1: Assignments not submitted per module

Late Submissions		
Module	Control (20)	Experimental (22)
1	0	1
2	1	2
3	1	3
4	1	2
5	1	5
6	4	5
7	2	4
Total	10	22

Table 2: Late assignments submitted per module

Mean Grade Point		
Module	Control	Experimental
1	71%(3.72)	75%(2.05)
2	79% (2.08)	71%(2.33)
3	73%(3.19)	73%(2.55)
4	62%(3.72)	66%(2.49)
5	74%(4.26)	75%(2.44)
6	67%(3.41)	67%(1.78)
7	56%(3.48)	65%(2.50)

Table 3: Mean grade points (with standard deviations) scored on the quiz by both groups

The data collected lays out some interesting points. The experimental group, at an anecdotal level, showed a greater inclination to submit the final assignment as compared to control group. Bear in mind that the experimental group students - by the time they submit the final assignment - have already submitted multiple assignments on the topic. A non-submission rate, that is almost half of the control group, may hint at the student's proclivity and willingness at submitting the final assignment. We believe that a better non-submission rate for the experimental group, even after doing multiple rounds of assignments is a healthy indicator of voluntary student engagement with the course.

Even though the non-submission rate is lower in the experimental group, the late submission rate is higher by over 100%. Late submissions in both group were allowed to see that if given the time, would students be motivated enough to work on the assignments? We found that students were more willing to work on the assignments in the experimental group even if that meant submitting it late. This is evident from the fact that there are more late submissions in experimental group than no submissions. The trend is reverse in the control group. This is to reiterate that the data presented here for experimental group is for the last cumulative assignment. By this time, for the same module, students would have submitted many incrementally difficult assignments, and a general student fatigue is expected which may speak for the higher number of late submissions.

Table 4 presents the end of module quiz grades for both groups. The groups were administered the exact same quizzes. There seems to no significant difference in the quiz performance for the groups, though the standard deviation in the experimental group seems to be on the lower side than that of the control group. Does that mean that constant practice, even though unable to improve overall group performance on quizzes, can help stem high variability of individual performance in the group? Could it be because weak students were able to improve their performance gradually? We cannot say anything for sure given such small sample size but the data does provide directions for potential explorations.
The groups were administered the exact same final exam. The two part exam consisted of writing a JAVA program and a multiple choice quiz that covered all seven modules. The JAVA program was worth two-third of the total points, and the quiz, one-third. Table 5 illustrates the data.

Group	Average Final Quiz Score	Average JAVA Program Score	Cumulative Average
Control	66%	51%	56%
Experi-mental	74%	71%	72%

Table 4: Final exam score for both groups

It is quite interesting to note that while there was no significant difference between module quiz scores, the experimental group performed much better in the final exam. Even though the gains in the final quiz are marginal, the experimental group outperformed the control group by 20% in JAVA program writing. The overall cumulative improvement in final exam mean score was 16%. These numbers may insinuate that–for the experimental group–the increased practice led to an improvement in final exam score, though it is too early to say anything with high degree of confidence due to such a small sample size. Nevertheless, the final exam numbers are encouraging.

	No Submissions							
Module	1	2	3	4	5	6	7	Total
1	0	0	0	0	-	-	-	0
2	0	0	0	0	0	0	-	0
3	0	0	0	0	-	-	-	0
4	0	1	0	1	0	0	-	2
5	0	0	0	0	1	-	-	1
6	0	0	1	0	1	1	3	6
7	0	2	1	1	3	-	-	7

Table 5: Assignments not submitted

Intra Group Analyses

Table 6 and 7 present detailed assignment submission data for the experimental group. The first column represents the module/chapter that was covered, and the numbered columns represent the assignment number in that particular module. Some modules had four, some five, and some had seven assignments. The instances of no submissions are relatively very low as compared to late submissions. Similar trend was missing in the control group.

	Late Submissions							
Module	1	2	3	4	5	6	7	Total
1	0	1	2	1	-	-	-	4
2	2	1	2	2	0	2	-	9
3	0	0	1	3	-	-	-	4
4	2	1	3	2	1	2	-	11
5	2	2	3	4	5	-	-	16
6	2	1	4	4	2	1	5	19
7	2	5	6	5	4	-	-	22

Table 6: Assignments submitted late

Table 8 presents a cumulative summary of the assignments. Cumulatively, only about 2% of the total assignments were not submitted. This could mean many things; one of the possible explanations might be that given the right conditions, the students were willing to engage more. Late submissions were allowed with reduced credit, and cumulative late submission rate stands at about 10.5%.

The instances of both late and no submissions increase as the course progresses, even though the rate of increase of no submissions is low as compared to late submissions. This may be explained by the fact that the concepts to be learned become complex as the course progresses, and some students might have given up on some of the assignments.

Module No	Maximum Possible Submissions	Not Submitted	Late Submissions
1	88	0	4
2	132	0	9
3	88	0	4
4	132	2	11
5	110	1	16
6	154	6	19
7	110	7	22
Total	814	16(1.9%)	85(10.5%)

Table 7: Assignment Summary

End of Course Survey

With the experimental group, we also conducted an end of the course survey to gauge how AAAD was received by our students. Participation was 100%. The questions were primarily centered around the potential impact of high number of assignments on their motivation, stress levels, and their choice between AAAD and the usual method of single assignment per module used at our department. The full survey is listed in appendix C. A few questions are discussed in the following paragraph.

Effectiveness of AAAD

One of the questions asked the students about how they felt about the utility and effectiveness of AAAD in completing the course satisfactorily. A surprising 90% of the students answered that they felt positive/better about using this technique while 10% reported in negative, and answered that they felt slightly worse.

Another question asked the students about the utility of doing a daily assignment in learning computer programming. A whopping 100% of the students felt that it is useful. This gives us some confidence that given the right cognitive load and environment, students do see potential value in constant practice for learning programming.

AAAD vs Normal Course Delivery

Another important question asked the students about their choice between AAAD and the normal course delivery mechanism of doing one assignment per week. 95% of the students said that they would prefer AAAD. Hence, the students overwhelmingly choose AAAD as a mode of course delivery over our normal delivery method. This, we believe, is a very important piece of feedback for us.

Impact of AAAD on Student Stress Levels

Another very important question on the survey asked the students about their perception of stress levels about doing so many assignments. Half of the students answered that AAAD made it easy for them to manage stress, 32% said it increased their stress levels, and 18% choose that it made no difference. We were initially concerned that a high percentage of students might report increased stress levels. Just 18% students choosing higher stress levels came as quite a surprise. If this indeed is the case, it is one of the big incentives for us to continue to utilize, and improve this technique further.

4. DISCUSSION

With such a small sample size, it is quite early to generalize the utility of this technique, but the initial results do reveal some interesting insights. Most of the students seem to find AAAD beneficial, even if it means spending more time than usual to work on so many assignments.

Potential Strengths

According to the assignment data collected and student responses on the survey, it is clear that most students show an inclination towards practicing more as long as the cognitive load is manageable. This becomes clear from the minimal no-submission and late-submission instances during module 1 to 5 that cover basic JAVA concepts. Module 6 and 7 cover complex concepts such as 2D arrays and file operations. The instances of no-submission and late-submission rise during these modules. For future research, we contemplate breaking down the assignments further in module 5 and 6, to see if that would reduce the instances of late and no submissions. Overall, this technique, appears to successfully increase student engagement in the course.

Another strength is the high degree of acceptance students showed towards this technique. It seems that students engaged in the course not just because they were pushed by daily deadlines; they seemed to have embraced the method, and found value in it. Even if they had to spend more time consistently doing assignments, they argued that it helped them learn programming, and positively pushed them to engage themselves with the course.

Potential Limitations

It is no doubt that the workload of this technique may be perceived as higher when compared to orthodox course delivery. The pressure of completing an assignment every day can still lead to student frustration, and may even exacerbate the very factor the technique was designed to mitigate. Results and responses, however, show that the technique successfully navigated these roadblocks. It remains to be seen, if these results can be replicated in future courses.

Another significant potential limitation of this technique is its resource intensiveness. Since students have do so many assignments, they tend to ask many more questions about the concepts, as well as clarifications on assignments. Providing timely feedback is challenging even when the instructor has a course grader. Grading so many assignments, in our experience, was one of the major concerns, as this may inadvertently lead to grading fatigue. Future research will investigate simulated software and automatic grading systems to reduce this grading workload.

Another important aspect of employing this technique was the continual and immediate presence of instructor and tutor support. Without this perennial support, this technique may be rendered less effective. Our experience in a more traditional approach is that about 50%-60% of the class asked questions on assignments on the day the assignments were due. Since students have a due date almost every day of the week, AAAD requires continuous tutor support due to sheer volume of the queries. If these questions remain unaddressed at the outset, it may cause learning gaps for the students. Since the subsequent assignments build on previous assignments, it may have a snowball effect, which is highly undesirable.

Another very important point of concern is that many of our students work full time. For them, as evidenced by comments in the survey, it is difficult to schedule time every day to finish the assignments. The peculiar observation, however, is that even the full time working students appreciated AAAD technique; it is just that they find it difficult to schedule assignment time every day.

5. CONCLUSION AND FUTURE WORK

Students in our introductory programming course agree that an assignment a day technique added value to their process of learning computer programming.

AAAD helped them practice consistently, thereby improving their enthusiasm about the course. Though there was no significant differences in the individual chapter quiz scores between the groups, the experimental group performed much better in the final exam.

Even though the students reported that they spent more time on the assignments, and had mixed reactions towards it, they overwhelmingly appreciated the value it brought to the table, and were convinced of its efficacy. The survey responses indicate that though the technique was very well received, it was not without its challenges. Firstly, grading a large number of assignments, and providing high volume of feedback is resource intensive. Continuous tutor support is also required to help stem student frustration, and to give them the feeling that help is always available.

Our future work includes finding ways to mitigate the load on the instructor, tutor/grader and students while maintaining the integrity of the technique, which is, continual practice and feedback. One aspect is the use of automatic grading systems to reduce the grading load. We also envisage coupling an automatic grading system with an artificial tutor bot capable of answering basic questions about the course, assignments, and programming simple concepts. Finally, we want to review the structure and design of the assignments to determine if there is a way to minimize questions. We are encouraged with this initial study and the promise of future research. We are contemplating using the same technique in our online programming course to see the technique's applicability in an online environment.

6. REFERENCES

Alexandron, G., Armoni, M., Gordon, M. & Harel, D. (2014). Scenario–based programming: Reducing the cognitive load, fostering abstract thinking. In Companion Proceedings of the 36th International Conference on Software Engineering (pp. 311–320). New York, NY: ACM.

Allan, V. H. & Kolesar, M. V. (1997). Teaching computer science: a problem solving approach that works. ACM SIGCUE Outlook, 25(1–2), 2–10.

Antonio Crego, María Carrillo-Diaz, Jason M. Armfield and Martín Romero Stress and Academic Performance in Dental Students: The Role of Coping Strategies and Examination-Related Self-Efficacy Journal of Dental Education February 2016, 80 (2) 165-172.

Beaubouef, T. B. & J. Mason (2005). Why the High Attrition Rate for Computer Science Students: Some Thoughts and Observations. Inroads – The SIGCSE Bulletin, 37(2), 103–106.

Bennedsen, J. & Caspersen, M. E. (2007). Failure rates in introductory programming. ACM SIGCSE Bulletin, 39(2), 32–36.

Brabrand, C. & Dahl, B. (2009). Using the SOLO Taxonomy to Analyze Competence Progression of University Science Curricula. Higher Education, 58(4), 531–549.

Brown, S. W., & Bennett, E. D. (2002). The role of practice and automaticity in temporal and nontemporal dual-task performance. Psychological Research, 66, 80–89.

Eckerdal, A. (2009). Novice Programming Students' Learning of Concepts and Practice. Doctoral dissertation, Acta Universitatis Upsaliensis.

Glover, J.A., Ronning, R.R. and Bruning, R.H.: 1990, Cognitive Psychology for Teachers, Macmillan, New York.

Howles, T. (2009). A study of attrition and the use of student learning communities in the computer science introductory programming sequence. Computer Science Education, 19(1), 1–13.

Kalchman, M., Moss, J., & Case, R. (2001). Psychological models for the development of mathematical understanding: Rational numbers and functions. In S. M. Carver & D. Klahr (Eds.), Cognition and instruction: Twenty-five years of progress (pp. 1-38). Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers.

Kaplan, D. S., Liu, R. X., & Kaplan, H. B (2005). School related stress in early adolescence and academic performance three years later: The conditional influence of self-expectations. Social Psychology of Education, 8, 3-17.

Kim, J. & Lerch, F. J. (1997). Why is programming (sometimes) so difficult? Programming as scientific discovery in multiple problem spaces. Information Systems Research, 8(1), 25–50.

Kinnunen, P. & Malmi, L. (2006). Why students drop out CS1 course?. In Proceedings of the Second International Workshop on Computing Education Research (pp. 97–108). New York, NY: ACM.

Lahtinen, E., Ala–Mutka, K. & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. ACM SIGCSE Bulletin, 37(3), 14–18).

Luxton-Reilly, A. (2016). Learning to program is easy. In Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (pp. 284–289). New York, NY: ACM.

Margulieux, L. E., Catrambone, R. & Schaeffer, L. M. (2018). Varying effects of subgoal labeled expository text in programming, chemistry, and statistics. Instructional Science, 10.1007/s11251-018-9451-7.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B., Laxer, C., Thomas, L., Utting, I. &, Wilusz, T. (2001). A Multi–national, Multi–institutional Study of Assessment of Programming Skills of First–year CS Students. ACM SIGCSE Bulletin, 33, 125–180.

Mendes, A. J., Paquete, L., Cardoso, A. & Gomes, A. (2012). Increasing student commitment in introductory programming learning. In Frontiers in Education Conference (FIE) (pp. 1–6). New York, NY: IEEE.

Newman, R., Gatward, R. & Poppleton, M. (1970). Paradigms for teaching computer programming in higher education. WIT Transactions on Information and Communication Technologies, 7, 299–305.

Oliver, D., Dobele, T., Greber, M. &Roberts, T. (2004). This Course has a Bloom Rating of 3.9. In Proceedings of the Sixth Australasian Conference on Computing Education (ACE ’04) (pp. 227–231). Darlinghurst, Australia: Australian Computer Society.

Rist, R. S. (1986). Plans in Programming: Definition, Demonstration, and Development. In Soloway, E. & Iyengar, S., eds., Empirical Studies of Programmers. Norwood, NJ: Ablex Publishing, pp. 28–47.

Rist, R. S. (1989). Schema Creation in Programming. Cognitive Science, 13, 389–414.

Rist, R. S. (1995). Program Structure and Design. Cognitive Science, 19, 507–562.

Rist, R. S. (2004). Learning to Program: Schema Creation, Application, and Evaluation. In Fincher, S. & Petre, M., eds., Computer Science Education Research. London, UK: Taylor & Francis, pp. 175–195.

Robins, A. V., Rountree, J. & Rountree, N. (2003). Learning and teaching programming: A review and discussion. Computer Science Education, 13(2), 137–172.

Rogalski J. & Samurçay R. (1990). Acquisition of programming knowledge and skills. In J. M. Hoc, T. R. G. Green, R. Samurçay & D. J. Gillmore, eds., Psychology of Programming. London: Academic Press, pp. 157–174.

Sheard, J. & Hagan, D. (1998). Our failing students: a study of a repeat group. ACM SIGCSE Bulletin, 30(3), 223–227.

Sorva, J. (2013). Notional machines and introductory programming education. ACM Transactions on Computing Education (TOCE), 13(2), Article 8 (31 pages).

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. Cognitive Science, 12(2), 257–285.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. Learning and Instruction, 4(4), 295–312.

Utting, I., Tew, A. E., McCracken, M., Thomas, L., Bouvier, D., Frye, R., Paterson, J., Caspersen, M., Kolikant, Y., Sorva, J. & Wilusz, T. (2013). A fresh look at novice programmers' performance and their teachers' expectations. In Proceedings of the ITICSE Working Group Reports Conference on Innovation and Technology in Computer Science Education (pp. 15–32). New York, NY: ACM.

Watson, C. & Li, F. W. (2014). Failure rates in introductory programming revisited. In Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education (pp. 39–44). New York, NY: ACM.

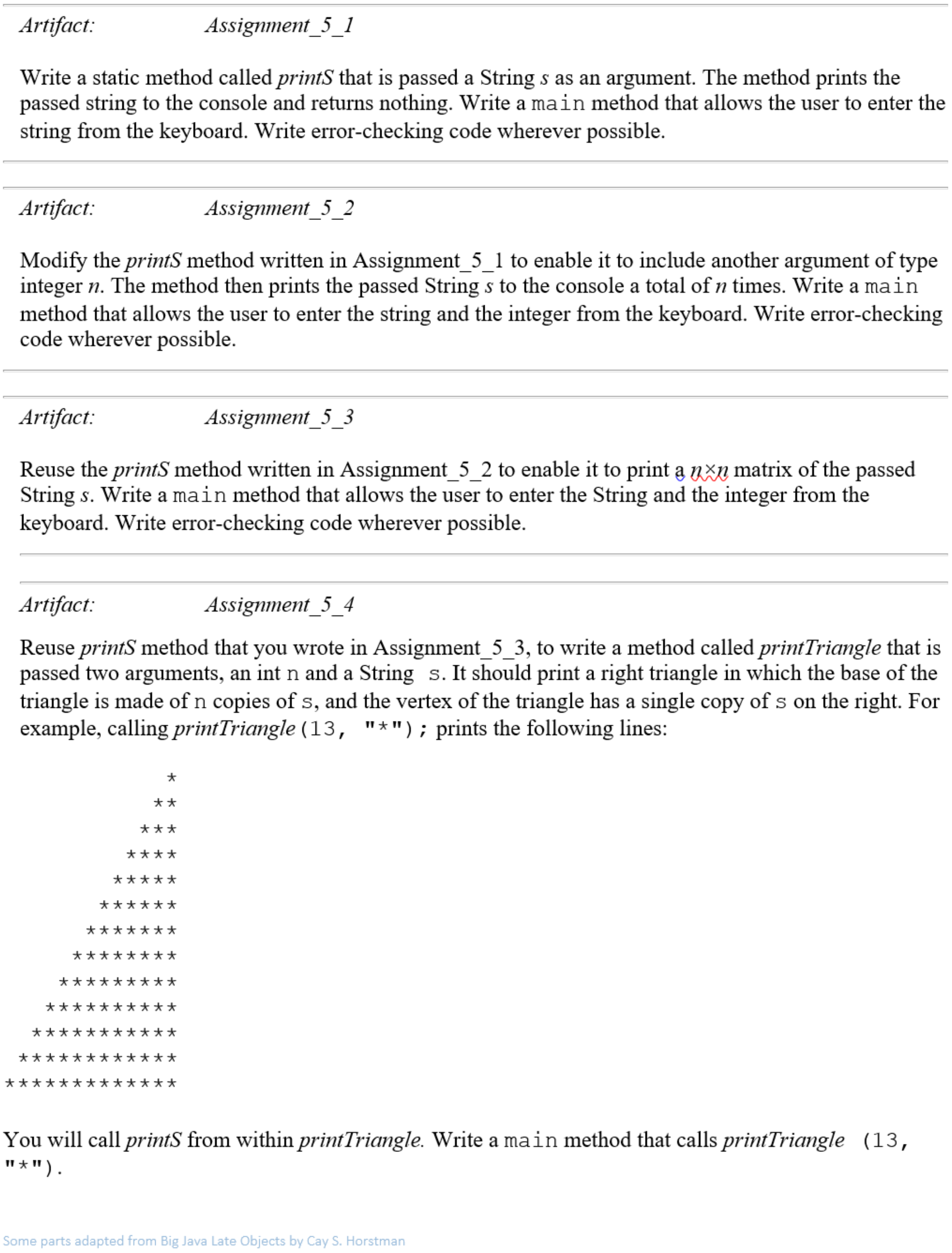
Winslow L E (1996) Programming pedagogy – A psychological overview. ACM SIGCSE Bulletin, 28(3), 17–22.

APPENDIX A

Table 8: Increment in congnitive load with time

Assignment No.	Description	Concepts Tested	Cognitive Load
1	Write a method printS that takes a string as an input and prints it to the console.	Rudimentary method writing.	Low
2	Modify the above method printS and enable it to take another argument, an integer, *n*. The method then prints the string *n* times in a line.	Method writing, method calling, method modification.	Low
3	Reuse printS to print a user entered string *n*×*n times* i.e a square with each element as the string	User input, loops, method writing, method calling	Medium
4	Reuse printS method to print a right angle triangle in terms of user entered string	User input, loops, method writing, method calling, Problem solving	Medium
5	Reuse printS to print a pyramid in terms of user entered string	User input, loops, method writing, method calling, Problem solving	High

APPENDIX B



Artifact: *Assignment_5_1*

Write a static method called *printS* that is passed a String *s* as an argument. The method prints the passed string to the console and returns nothing. Write a `main` method that allows the user to enter the string from the keyboard. Write error-checking code wherever possible.

Artifact: *Assignment_5_2*

Modify the *printS* method written in Assignment_5_1 to enable it to include another argument of type integer *n*. The method then prints the passed String *s* to the console a total of *n* times. Write a `main` method that allows the user to enter the string and the integer from the keyboard. Write error-checking code wherever possible.

Artifact: *Assignment_5_3*

Reuse the *printS* method written in Assignment_5_2 to enable it to print a $n \times n$ matrix of the passed String *s*. Write a `main` method that allows the user to enter the String and the integer from the keyboard. Write error-checking code wherever possible.

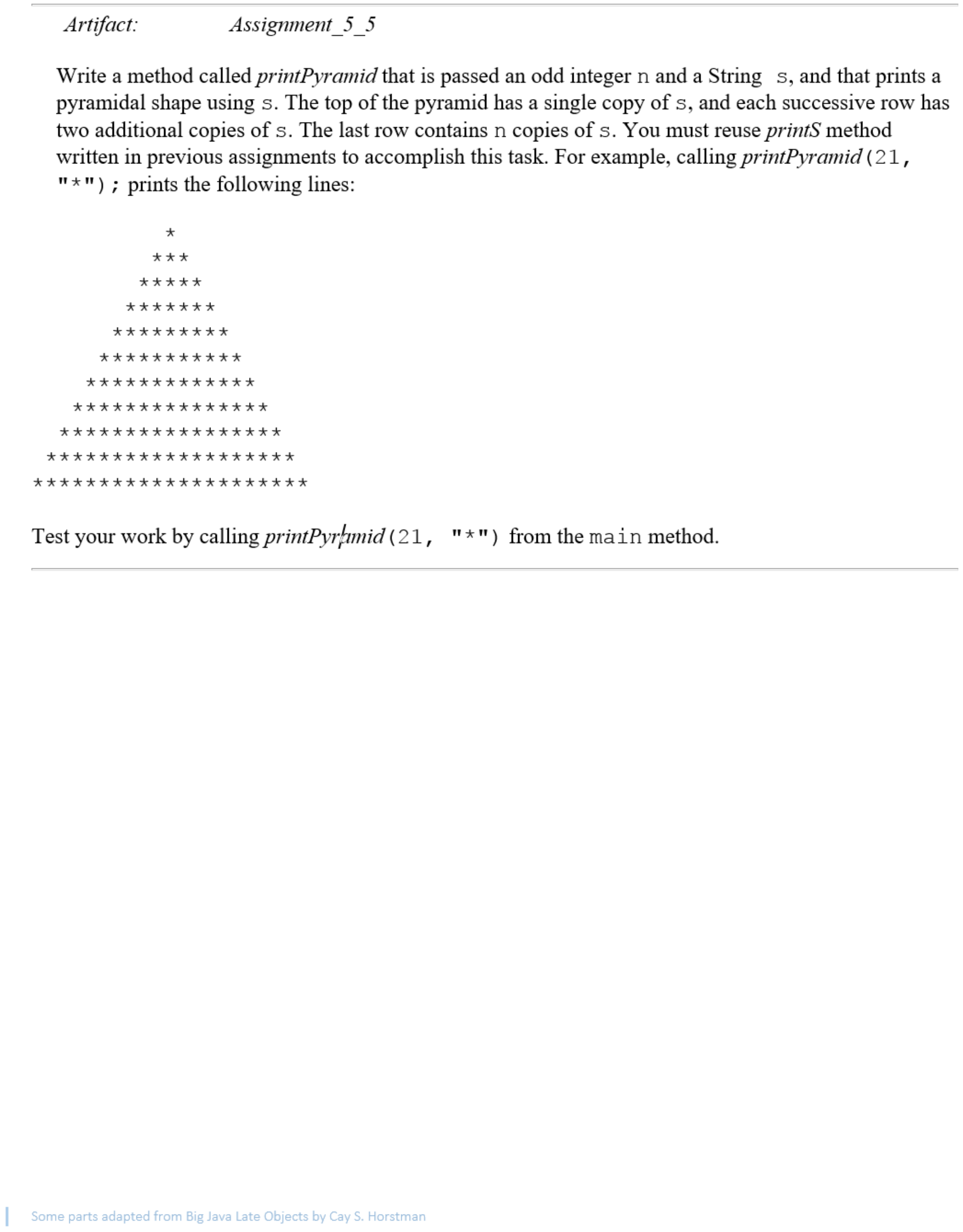
Artifact: *Assignment_5_4*

Reuse *printS* method that you wrote in Assignment_5_3, to write a method called *printTriangle* that is passed two arguments, an int `n` and a String `s`. It should print a right triangle in which the base of the triangle is made of `n` copies of `s`, and the vertex of the triangle has a single copy of `s` on the right. For example, calling *printTriangle* `(13, "*");` prints the following lines:

```
            *
           **
          ***
         ****
        *****
       ******
      *******
     ********
    *********
   **********
  ***********
 ************
*************
```

You will call *printS* from within *printTriangle.* Write a `main` method that calls *printTriangle* `(13, "*").`

Some parts adapted from Big Java Late Objects by Cay S. Horstman



Artifact: *Assignment_5_5*

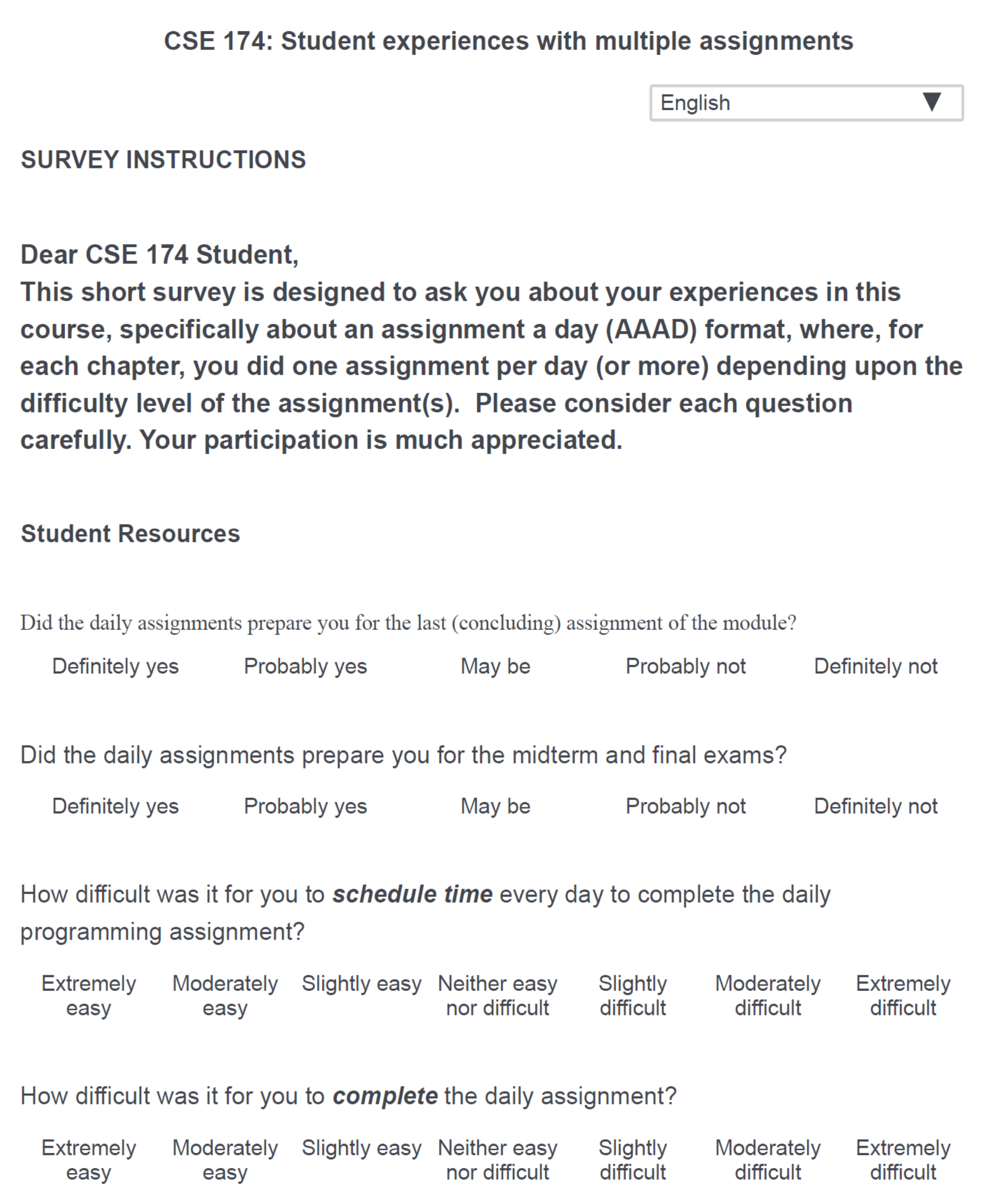
Write a method called *printPyramid* that is passed an odd integer `n` and a String `s`, and that prints a pyramidal shape using `s`. The top of the pyramid has a single copy of `s`, and each successive row has two additional copies of `s`. The last row contains `n` copies of `s`. You must reuse *printS* method written in previous assignments to accomplish this task. For example, calling *printPyramid*`(21, "*");` prints the following lines:

```
          *
         ***
        *****
       *******
      *********
     ***********
    *************
   ***************
  *****************
 *******************
*********************
```

Test your work by calling *printPyramid*`(21, "*")` from the `main` method.

Some parts adapted from Big Java Late Objects by Cay S. Horstman

APPENDIX C



CSE 174: Student experiences with multiple assignments

English ▼

SURVEY INSTRUCTIONS

Dear CSE 174 Student,
This short survey is designed to ask you about your experiences in this course, specifically about an assignment a day (AAAD) format, where, for each chapter, you did one assignment per day (or more) depending upon the difficulty level of the assignment(s). Please consider each question carefully. Your participation is much appreciated.

Student Resources

Did the daily assignments prepare you for the last (concluding) assignment of the module?

Definitely yes	Probably yes	May be	Probably not	Definitely not

Did the daily assignments prepare you for the midterm and final exams?

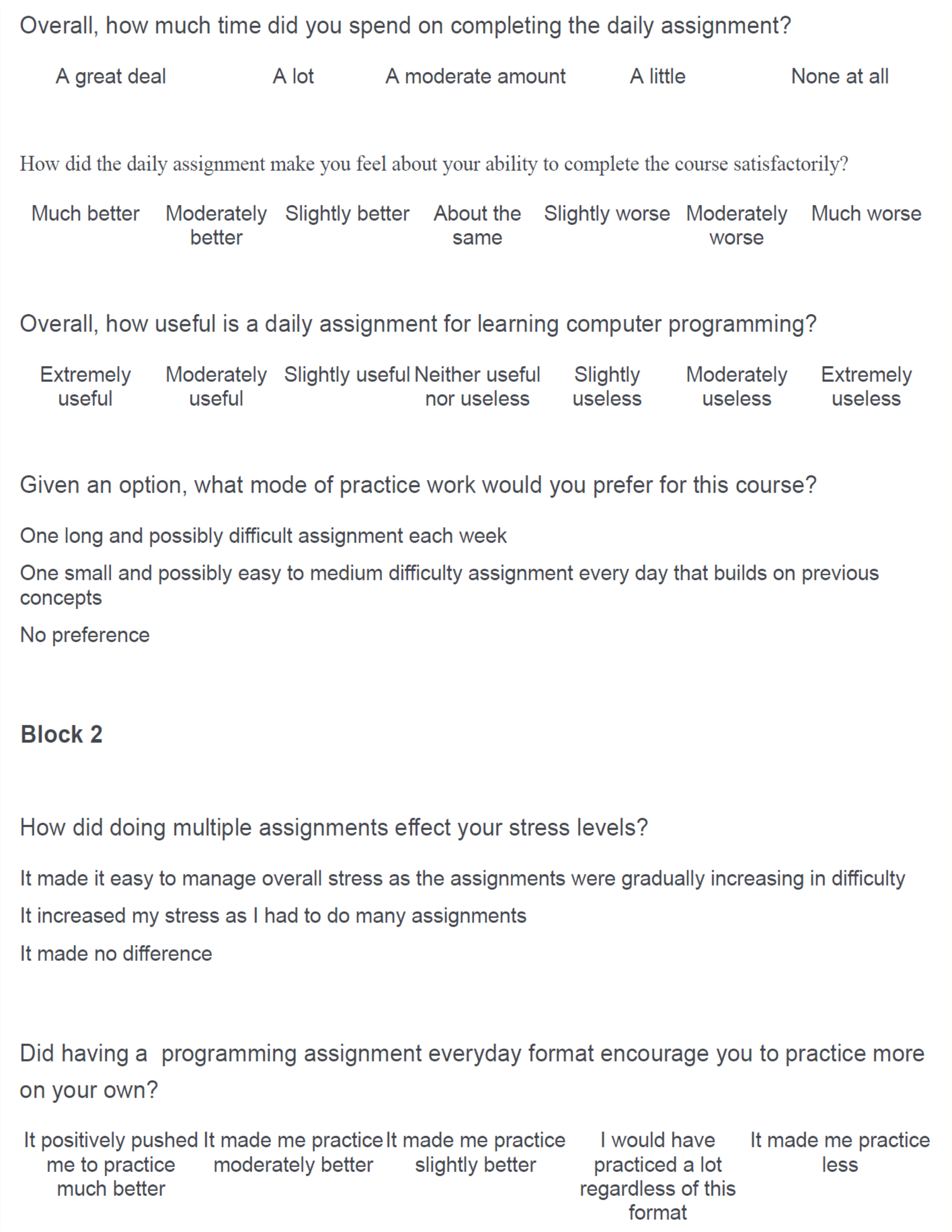
Definitely yes	Probably yes	May be	Probably not	Definitely not

How difficult was it for you to ***schedule time*** every day to complete the daily programming assignment?

Extremely easy	Moderately easy	Slightly easy	Neither easy nor difficult	Slightly difficult	Moderately difficult	Extremely difficult

How difficult was it for you to ***complete*** the daily assignment?

Extremely easy	Moderately easy	Slightly easy	Neither easy nor difficult	Slightly difficult	Moderately difficult	Extremely difficult



Overall, how much time did you spend on completing the daily assignment?

A great deal	A lot	A moderate amount	A little	None at all

How did the daily assignment make you feel about your ability to complete the course satisfactorily?

Much better	Moderately better	Slightly better	About the same	Slightly worse	Moderately worse	Much worse

Overall, how useful is a daily assignment for learning computer programming?

Extremely useful	Moderately useful	Slightly useful	Neither useful nor useless	Slightly useless	Moderately useless	Extremely useless

Given an option, what mode of practice work would you prefer for this course?

One long and possibly difficult assignment each week

One small and possibly easy to medium difficulty assignment every day that builds on previous concepts

No preference

Block 2

How did doing multiple assignments effect your stress levels?

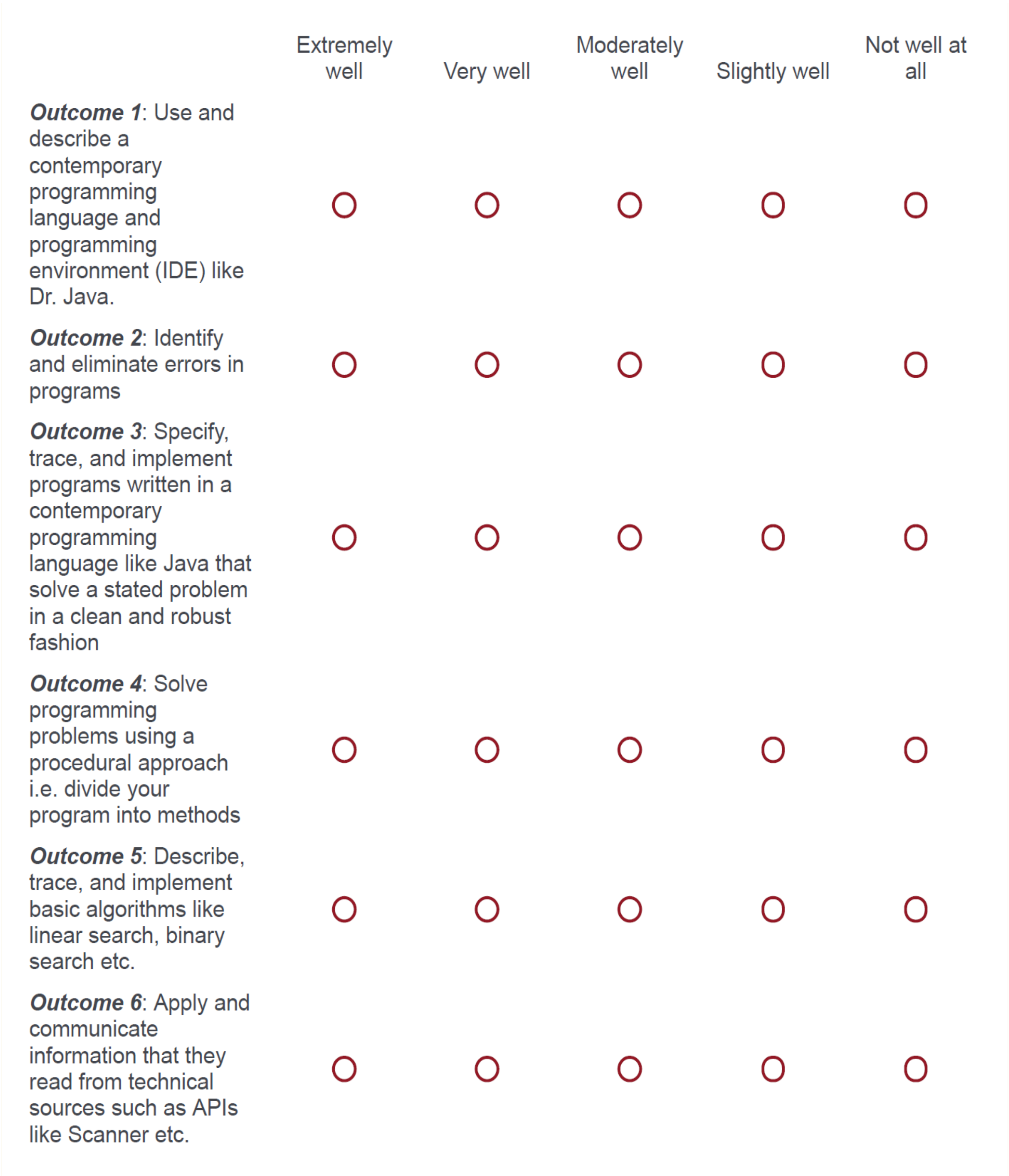
It made it easy to manage overall stress as the assignments were gradually increasing in difficulty

It increased my stress as I had to do many assignments

It made no difference

Did having a programming assignment everyday format encourage you to practice more on your own?

It positively pushed me to practice much better	It made me practice moderately better	It made me practice slightly better	I would have practiced a lot regardless of this format	It made me practice less



	Extremely well	Very well	Moderately well	Slightly well	Not well at all
Outcome 1: Use and describe a contemporary programming language and programming environment (IDE) like Dr. Java.	○	○	○	○	○
Outcome 2: Identify and eliminate errors in programs	○	○	○	○	○
Outcome 3: Specify, trace, and implement programs written in a contemporary programming language like Java that solve a stated problem in a clean and robust fashion	○	○	○	○	○
Outcome 4: Solve programming problems using a procedural approach i.e. divide your program into methods	○	○	○	○	○
Outcome 5: Describe, trace, and implement basic algorithms like linear search, binary search etc.	○	○	○	○	○
Outcome 6: Apply and communicate information that they read from technical sources such as APIs like Scanner etc.	○	○	○	○	○

Learning How to Teach: The Case for Faculty Learning Communities

David Gomillion
dgomillion@mays.tamu.edu

Aaron Becker
abecker@mays.tamu.edu

Jordana George
jgeorge@mays.tamu.edu

Michael Scialdone
mscialdone@mays.tamu.edu

Information and Operations Management
Mays Business School
Texas A&M University
College Station, Texas

Abstract

Faculty learning communities, a specialized form of communities of practice, are not new. These communities provide opportunities for learning, feedback, and collegiality. Even with all of these benefits, many faculty have never participated in a learning community, sometimes because colleges and schools have not yet established one. This paper presents two cases in which faculty participated in a Faculty Learning Community and provides some recommendations for establishing a new community.

Keywords: Faculty Learning Communities, Communities of Practice, Scholarship of Teaching, Continuing Education

1. INTRODUCTION

Faculty learning communities are a specialized form of communities of practice. Wenger (2011, p. 1) defines communities of practice as “groups of people who share a concern or a passion for something they do an learn how to do it better as they interact regularly.” Not all communities of practice will be purpose-built to encourage learning; however, faculty learning communities have learning as a stated goal.

Many of those teaching at colleges and universities have no formal teaching education. Ironically, the famously underpaid K-12 educators have more education about educating minds than those in the Academy. While not universally true, teaching at some research universities (where most PhD degrees are awarded) may be viewed as a necessary evil, a task that is done to pay the bills, or worse yet, a distraction from the important research. As such, prospective faculty members quickly learn to do only the bare minimum when it comes to developing and delivering courses. When given another professor’s syllabus/course materials to work with, the assumption is that they will not be making any substantive changes. They have just been saved from a new class prep.

It is little wonder that in organizations where research is celebrated and teaching is tolerated that formal education in how people learn is omitted in the curriculum. The incentives strongly support publication in top-tier journals but only require teaching to meet some minimum threshold. As such, those who want to be excellent teachers are sometimes left ‘in the dark’ when it comes to how to improve their teaching.

There are many good resources for learning to be more effective in teaching. Scholars in education create articles, books, provide conference presentations, and often share their research online to try to help improve the state of the art of teaching. But faculty that are constantly prepping courses, delivering those courses, grading papers and projects, and creating exams often find themselves with precious little time to keep current. And it can be lonely, hiding in an office and reading the current research in education.

Faculty may also struggle with how to solicit feedback. Universities have instructional design professionals, but they may serve an entire college or university. Most universities also have some type of instructional support system, but once again, it can be general in its scope. Feedback from peers can provide insight into what works and what does not. Bouncing ideas off colleagues provides synergistic learning – both parties think differently after the exchange. However, asking busy colleagues to sacrifice time to observe teaching or provide feedback on assignments can be an uncomfortable experience.

The academy values collegiality. Most promotion processes cite collegiality as necessary to continue employment. Working in a collegial environment is great. However, our work as academics tends to isolate us, each in our own classroom when teaching or office when researching. Further, our work environment might not currently be supportive.

One technique to keep current on educational research, obtain feedback, and increase collegiality is with a faculty learning community. Cox (2004, p. 8) defines a Faculty Learning Community as “a cross-disciplinary faculty and staff group of six to fifteen members … who engage in an active, collaborative, yearlong program with a curriculum about enhancing teaching and learning with frequent seminars and activities that provide learning, development, the scholarship of teaching, and community building.” This definition is more prescriptive than how it is intended in this paper. Layne et al. (2002) takes a more flexible approach to the activities and instead focuses on the sustained nature of the interaction, either a semester or an academic year. This contrasts with the typical professional development opportunities such as workshops and brownbag discussions that present one particular tool or technique.

This paper provides a brief overview of the research into faculty learning communities. It provides the experience of two faculty members that participated in different faculty learning communities. Next, it provides some suggestions and resources for establishing a faculty learning community.

2. THEORETICAL SUPPORT

To understand why faculty learning communities are useful constructs, this article will explore theoretical support for communities of practice in general and faculty learning communities in particular.

Communities of Practice

Organizations are successful insofar as they have the necessary resources to accomplish their work (Peteraf, 1993; Wernerfelt, 1984). Work within organizations have changed significantly because organizational knowledge is the most valuable asset (Grant, 1996, 2002). Thus, the most important work an organization can do is to generate new knowledge. This poses a problem for managers because knowledge is largely invisible.

Some type of organizational structure is needed to facilitate building and sharing knowledge. Valuable knowledge is often tacit, meaning people don’t know they know it, and if they do know they possess it, they have a hard time describing it or how they came to know it (Nonaka, 1994; Reber, 1989). As such, just writing it down can be difficult; yet, such knowledge is invaluable for groups to be able to innovate (Leonard & Sensiper, 1998). How do we share knowledge we do not know we have, or cannot put into words? This is where storytelling, apprenticeship, and communities help (Mládková, 2012).

A Community of Practice (CoP) has been defined as "a flexible group of professionals, informally bound by common interests, who interact through interdependent tasks guided by a common purpose thereby embodying a store of common knowledge" (as quoted from (Jubert, 1999, p. 166) in Davenport & Hall (2002, p. 171)). Current understandings of CoP draws from the situaFtion learning, distributed cognition, and communication studies (Davenport & Hall, 2002).

A CoP does not have to be co-located, in the same organization, or even in the same industry (Davenport & Hall, 2002). They simply share some common attribute. For instance, if all of the network engineers in town meet at a bar on Tuesday nights, they can be forming a community of practice. War stories of network bugaboos will be swapped, and everyone will increase in their knowledge. Tacit knowledge, like how to troubleshoot such wicked problems, will spread between members,and across organizational boundaries. But such communities could occur on forums just as easily.

The key benefits of a community of practice is to "radically galvanize knowledge sharing, learning, and change" (E. C. Wenger & Snyder, 2000, p. 139). Thus, organizations should nurture CoPs to help them be more competitive, such as when they need to drive strategy or start new lines of business. Some observed benefits include solving problems quickly, transferring best practices, developing professional skills, and helping organizations recruit and retain the human resources that they need (E. C. Wenger & Snyder, 2000).

Faculty Learning Communities

A specialized form of CoP is the faculty learning community (FLC). In a FLC, participants gather regularly to discuss how to teach generally, sometimes with a prescribed resource, but usually for a sustained period and with participants from different disciplines (Cox, 2004; Layne et al., 2002).

The FLC has been a topic of interest since the Carnegie Foundation's Scholarship Reconsidered (Boyer, 1990) report emphasized that the scholarship of teaching has been neglected in favor of basic scientific inquiry (Richlin & Cox, 2004). As participants in this conference know, scholarly teaching (using data insights to improve our course) and the scholarship of teaching (publishing new models based on the insights we have gained) provides significant value. But this is new to much of the Academy, and FLC can be a mechanism to help spread the message of scholarship of teaching (Richlin & Cox, 2004).

In addition to evangelizing the scholarship of teaching, FLCs can help provide feedback to faculty members (Cox, 1999). The typical mentoring relationship is one-on-one, where a person asks a question and someone with different experience provides guidance to help that person improve. Not only is this a great way to work; it is a form of apprenticeship. But if we expand the circle to more than just a dyad, more opinions can be sought, and more people can learn from the exchange. The mentor is just as likely to learn from other members of the community as anyone else. So FLCs can be a mechanism to help provide peer feedback.

The third major advantage discussed in the literature about FLCs is breaking down barriers between faculty (Cox, 2004). It is easy for faculty members to feel isolated; in fact, a senior scholar warned one author that being a professor was a "lonely life" as he was applying for a PhD program. Teaching is done with students, yes, but very little peer interaction. Grading is done in a largely solitary situation. Preparing for class is likewise done alone. And much of research is completed alone, even when we will pass a draft of a paper along to a co-author. FLC creates a regularly scheduled opportunity to gain that human connection that is so easily lost.

3. EXPERIENCES IN LEARNING COMMUNITIES

One of the authors experienced a faculty learning community based on a strategic vision for what the business school needed students to know to be successful. The school had created a new plan for how to imbue these characteristics with all of the likely candidates: critical thinking, acting ethically, leading, and communicating to name a few. But the question was how to operationalize these core competencies. To explore this, the business school formed a new faculty learning community, with membership open to volunteers across the departments. Faculty striving to be better teachers self-selected into the community.

The community came together and discussed goals and why we had volunteered to take part in the bi-weekly meetings. This helped build true community as we got to know each other. We set the book we would use to guide some discussion, Paul Hanstedt's *Creating Wicked*

Students. This was discussed, with each faculty member bringing in other resources. But as the community read and discussed the topics, members also created, recreated, or updated an assignment to apply the vision, use the ideas from the community, and measure success in teaching one of the major core competencies. The community lasted for the full academic year. To encourage continued participating, members of the community that persisted throughout the entire year were awarded an additional grant for teaching materials or professional development.

Another of the authors experienced a different faculty learning community with a broader goal: to get its members engaged with the Scholarship of Teaching and Learning (SoTL) community. That engagement included both reading extant literature concerning the problems that the group members were facing in their respective classes, and trying novel approaches to solving those problems in order to ultimately publish research in that area and thus further the scholarship.

As with the first community, there was a book serving to guide our discussions (in this case it was *Inquiry into the College Classroom: A Journey towards Scholarly Teaching* by Paul Savory, Amy Nelson Burnett, and Amy Goodburn). However, whereas the formal goal of the first community was to redesign a single assignment, the formal goal of this community was to redesign an entire course. The structure of the book aided this redesign process, as the chapters laid out a sequence each member could follow.

Additionally, the faculty member who started the group (being well versed in the SoTL literature) served as a mentor to each of the members, often bringing research to their attention that was directly applicable to the sorts of problems that they were trying to solve in the redesign of their classes. This was often an eye-opening experience for members, discovering that others had encountered the same problems as them and had developed various means of addressing those issues.

As with the first community, the group was defined for a specific period. Initially, it was intended to last only a semester. However, because the group members enjoyed the interactions and the course redesign process took longer than expected, the group ended up meeting for an entire academic year. As with the other community, members were awarded a bursary for participation in the group for its duration. The monetary amounts were not large, but the members still appreciated that their efforts were supported by the college.

Both communities were formed under the auspices of formal goals. But in each, what was gained was far more: there was a sense of community, a group of peers to provide feedback on how to engage students, and problem-solving for issues each faced in the classroom. It became a support group, a sounding board, and an expert exchange all in one. Moreover, as others voiced their problems and heard how some other member solved a similar problem, the learning was shared beyond the typical one-on-one mentoring approach. In many ways, teaching can feel like a very solitary exercise, in which our successes and failures are our own. These communities served to remind each of the authors that it does not have to be this way.

4. ESTABLISHING A NEW FACULTY LEARNING COMMUNITY

Cox (2004) provides a summary of the suggestions on establishing FLCs, chiefly in Appendices A and B. There are two major aspects: establishing community and architecting the FLC.

Establishing community means more than just scheduling a recurring meeting. Community is defined as a “feeling that members have belonging, a feeling that members matter to one another and to the group, and a shared faith the members’ needs will be met through their commitment to be together” (McMillan & Chavis, 1986, p. 9). The sense of belonging to a community is a psychological construct. That is felt by the members of the group. To create this kind of community, Cox (2004) highlights safety and trust, openness, respect, responsiveness, collaboration, relevance, challenge, enjoyment, esprit de corps, and empowerment. Each of these is necessary but not sufficient for community; the sense of belonging and membership cannot occur unless all of these are part of the culture of the group formed.

Learning communities do not have to be face-to-face. Palloff & Pratt (1999) provides guidance for how to bring that same sense of community in computer-mediated communications. While the article does not directly address CoPs, the advice on how to build community behaviors within the online course could be adapted to help a FLC flourish in an online forum.

At a slightly more tactical level, Cox (2004) recommends that FLCs be established with a mission and purpose, a curriculum, clear administration purposes and qualities, connections, affiliated participants, meetings and activities, scholarly process, assessment, and enablers and rewards.

In addition to the FLC-specific resources mentioned, since FLCs are specialized CoPs, the general CoP advice, such as that found in Shapiro & Levine (1999) can be helpful. Table 1 summarizes recommendations for establishing a new FLC.

Table 9. Building a New FLC

Organize a small group of champions	Find a few passionate like-minded people. Too many founders will make decisions difficult.
Identify the mission	Clearly state the goals of the group.
Put building blocks in place for culture	Decide in advance what type of culture is desired and which activities promote or degrade it.
Identify the scope of the community	Will the community serve one department or school or the entire university?
Identify potential members	Know your audience and take their needs into account.
Set up a community online platform	Will resources be fully open or only to group members? How will editing rights be managed?
Make joining easy	Engage in marketing the community, make membership as easy as possible.
Keep things current	Many communities die because activities go stale.
Understand and anticipate the 90-9-1 rule	In online communities, 90% of people lurk, 9% have some minimal level of interaction, and 1% will be proactive, providing the most content and participation.

5. CONCLUSION

Regardless of which processes might be used to establish it or the myriad idiosyncratic structures under which it might operate, a FLC is simply a group of faculty that come together regularly in a sustained effort to try to improve how they teach. Any structure you choose can still help improve the knowledge of scholarship of teaching, provide peer feedback, and help faculty members feel less isolated and more part of a community. Improving our teaching should be a goal that we all share.

Successful learning communities should have enough structure to encourage collaborative discussions. Participation in FLCs can lead to unexpected results, and time limits may be exceeded if (or when) participants find the interactions to be helpful

Learning communities can be face-to-face or they can be mediated by technology. But no matter how you structure it, what your stated goals are, or how you connect, FLCs are an excellent and rewarding way to help faculty enhance their craft and to stay more connected with collegues. So what are you waiting for? Go start one today!

5. REFERENCES

Boyer, E. L. (1990). *Scholarship Reconsidered: Priorities of the Professoriate*. Retrieved from https://eric.ed.gov/?id=ED326149

Cox, M. D. (1999). Peer Consultation and Faculty Learning Communities. *New Directions for Teaching and Learning*, *1999*(79), 39–49. https://doi.org/10.1002/tl.7905

Cox, M. D. (2004). Introduction to faculty learning communities. *New Directions for Teaching and Learning*, *2004*(97), 5–23.

Davenport, E., & Hall, H. (2002). Organizational knowledge and communities of practice. *Annual Review of Information Science and Technology*, *36*(1), 170–227.

Grant, R. M. (1996). Toward a knowledge-based theory of the firm. *Strategic Management Journal*, *17*(S2), 109–122.

Grant, R. M. (2002). The knowledge-based view of the firm. *The Strategic Management of Intellectual Capital and Organizational Knowledge*, *17*(2), 133–148.

Jubert, A. (1999). Developing an infrastructure for communities of practice: the Siemens experience. *ONLINE INFORMATION-INTERNATIONAL MEETING-*, 165–168. LEARNED INFORMATION LTD.

Layne, J., Froyd, J., Morgan, J., & Kenimer, A. (2002). Faculty Learning Communities. *32nd Annual Frontiers in Education*, *2*, F1A-F1A. https://doi.org/10.1109/FIE.2002.1158114

Leonard, D., & Sensiper, S. (1998). The role of tacit knowledge in group innovation. *California Management Review*, *40*(3), 112–132.

McMillan, D. W., & Chavis, D. M. (1986). Sense of community: A definition and theory. *Journal of Community Psychology*, *14*(1), 6–23.

Mládková, L. (2012). Sharing tacit knowledge within organizations: Evidence from the Czech Republic. *Global Journal of Business Research*, *6*(2), 105–115.

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science*, *5*(1), 14–37.

Palloff, R. M., & Pratt, K. (1999). *Building learning communities in cyberspace* (Vol. 12). San Francisco: Jossey-Bass.

Peteraf, M. A. (1993). The cornerstones of competitive advantage: a resource-based view. *Strategic Management Journal*, *14*(3), 179–191.

Reber, A. S. (1989). Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General*, *118*(3), 219.

Richlin, L., & Cox, M. D. (2004). Developing scholarly teaching and the scholarship of teaching and learning through faculty learning communities. *New Directions for Teaching and Learning*, *2004*(97), 127–135.

Shapiro, N. S., & Levine, J. H. (1999). *Creating Learning Communities: A Practical Guide to Winning Support, Organizing for Change, and Implementing Programs. Jossey-Bass Higher and Adult Education Series.* ERIC.

Wenger, E. (2011). *Communities of practice: A brief introduction*.

Wenger, E. C., & Snyder, W. M. (2000). Communities of practice: The organizational frontier. *Harvard Business Review*, *78*(1), 139–146.

Wernerfelt, B. (1984). A resource-based view of the firm. *Strategic Management Journal*, *5*(2), 171–180.