INFORMATION SYSTEMS EDUCATION JOURNAL

Volume 17, No. 4
August 2019
ISSN: 1545-679X

In this issue:

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 AITP-EDSIG who perform the editorial and review processes for ISEDJ.

INFORMATION SYSTEMS EDUCATION JOURNAL

Easy as Py: A First Course in Python with a Taste of Data Analytics

Mark Frydenberg
mfrydenberg@bentley.edu

Jennifer Xu
jxu@bentley.edu

Computer Information Systems Department
Bentley University
Waltham, MA 02452

Abstract

Python is a popular, general purpose programming language that is gaining wide adoption in beginning programming courses. This paper describes the development and implementation of an introductory Python course at a business university open to students in a variety of majors and minors. Given the growing number of career opportunities in analytics, the instructors felt that including a module on Data Analytics would add relevance and interest in the course. A survey given at the end of the semester shows students found this topic to be relevant to their future uses of Python. The paper also discusses challenges in teaching a first programming course to students with varying levels of programming experience.

Keywords: Python, Data Analytics, programming languages, programming courses

1. INTRODUCTION

Python is a first programming language in many secondary schools and universities (Siegfried, Siegfried, & Alexandro, 2016), and industry has embraced it as well. A recent study (Cass, 2017) showed that Python was the #1 language among top US Computer Science departments for teaching introductory programming courses (Guo, 2014) Companies such as Google, Facebook, Instagram, Netflix, and Dropbox all make use of Python because of its simplicity and ease of deployment and code maintenance, and variety of available code libraries. These features allow developers to focus on writing code for their specialized applications. (Reynolds, 2018)

In addition to Python's popularity in creating applications for the Internet of Things (Frydenberg, 2017) and bioinformatics (Kortsarts, Morris, & Utell, Janine, 2010), recent years have seen a major increase in data science and analytics applications implemented in Python.

Python in the Business Curriculum

While many engineering and information science departments offer Python in an engineering context, Python in the business curriculum serves Computer Information Systems (CIS) and Information Technology (IT) majors, minors, and students in other business programs by giving them the opportunity to learn a popular language that is widely used in business.

Appendix 1, Table 1 shows the list of the top 12 MIS programs (US News & World Report, 2018) and based on information in the course catalogs on their websites as of June 1, 2018, which schools require Python, and which offer it as an elective for Information Systems majors. Only MIT requires Python for business analytics majors, and only Indiana University Bloomington offers it as an elective. Many of these schools have engineering, computer science, or information science departments that offer a Python elective, but their course design likely focuses on programming concepts and algorithms rather than business applications. As business universities increase offerings in data science, analytics, and related fields, an understanding of Python will be essential.

Bentley University, a business university in Massachusetts, offered one section of Python as an experimental elective through its CIS Department for the first time during the fall 2017 semester, and three sections taught by two instructors during the spring 2018 semester. The instructor teaching the two spring sections sat in on the fall course so that both instructors could have a common basis of experience when modifying and preparing the course for the spring semester. Graded assignments and exams were the same across all three sections in the spring.

This paper explores considerations, assignments, and results of the implementation of an introductory Python course in a business school curriculum as well as student reactions to learning the language. The course fulfilled an Arts and Science elective.

The following questions guided this study:

- What factors motivate students to take (and universities to offer) Python as part of the business curriculum?
- In addition to programming fundamentals, which topics and applications of Python are considered relevant in a business context?
- What value do business students receive from taking an introductory Python course?

2. DESIGNING A PYTHON COURSE FOR BUSINESS STUDENTS

Undergraduate CIS majors at Bentley University are required to take a semester course in Java programming. Adding another language as an elective provides additional opportunities for students to strengthen their development skills, which will open more employment opportunities. Offering Python as an elective for CIS minors provides similar benefits.

Because Python is a very popular and widely used language in data-intensive disciplines, the course is also beneficial to Mathematical Sciences majors and minors. The introductory Python course has no prerequisites other than IT 101, a required course on digital literacy, covering technology concepts, Excel, and designing basic web pages with HTML, that is typically taken during the students' first year.

Topics

This course presented topics found in most introductory Python programming courses (McMaster, Sambasivam, Rague, & Wolthuis, 2017; Topi et al., 2010) , including:

- Variables, Data Types, and Expressions
- Loops and Selection Statements
- Strings and text files
- Lists and Dictionaries
- Functions
- Classes and Objects

Several introductory Python textbooks (Downey, 2016; Lambert, 2018) also offer chapters presenting applications that include graphics processing and user interface development. In planning the spring 2018 course, the instructors recognized that teaching even the most basic applications of data analytics would be of interest to business students. To accommodate this additional topic, the instructors chose to replace the chapters on graphics processing and user interface development with a module introducing the basic data analytics capabilities of Python.

As this is an introductory course, the instructors omitted advanced topics such as higher order functions (map, reduce, filter, lambda), inheritance and polymorphism, even though they were covered in the course textbook (Lambert, 2018).

Course Structure

The Python course met for two 80-minute sessions each week during the Fall 2017 and Spring 2018 semesters. Each class session included instructor-led demonstrations, lectures or presentations, and often, short, in-class exercises that reinforced the topics being presented. Students relied on instructor office hours and assistance from tutors in the University's CIS learning center when they needed assistance.

The course assignments included a standardized midterm and final exam, comprised of multiple choice, trace the output, fill in the missing code statements, and coding questions. In addition to short coding problems completed in class which counted toward their class participation scores, students completed seven major programming assignments for homework, as summarized in Appendix 1, Table 2.

3. INTRODUCING DATA ANALYTICS IN A FIRST PYTHON COURSE

Given the widespread use and highly promoted applicability of Python to data analytics, the instructors added a unit on this topic during the Spring 2018 semester. Even though most introductory Python text books do not include this content, accomplishing basic data analytics tasks are within reach of beginning programming students.

To use the pandas (an acronym derived from "Python and data analysis"), matplotlib (2D plotting) and numpy (scientific computing and numerical analysis) libraries for data analytics in a Python application, an understanding of objects and collections is useful. Providing a taste of data analytics in an introductory Python course becomes possible, as it builds on earlier topics and makes the course content more relevant for students.

Basic Data Analytics Capabilities with pandas, matplotlib, and numpy
The instructors spent three class sessions on data analytics. Much of the first session was spent having students install pandas, matplotlib and numpy libraries using the Miniconda distribution of Python (Continuum Analytics, 2018) onto their laptops. Miniconda provides a minimal Python installation containing additional libraries and packages. Even with step-by-step instructions, some students found the install process to be cumbersome. Students running macOS were challenged by the need to run installation commands in a bash shell, as they were not familiar with using a text-based command line interface to interact with an operating system.

During the first and second classes on Data Analytics, the instructors demonstrated programs that use the pandas DataFrame. A DataFrame is a two-dimensional data structure with predefined methods to sort, filter, and rearrange columns, create pivot tables, and perform other calculations and operations. Examples included reading data from a file and storing it to a DataFrame, printing data with and without column headings (and contrasting how to accomplish the same task without using the pandas module), sorting data by one or more columns, finding maximum and minimum values. After having written for loops to iterate through lists and dictionaries and a Grid class (Lambert, 2018, p. 330) to process data earlier in the course, students found interacting with a pandas DataFrame to be much more intuitive for processing two-dimensional data.

The instructors also demonstrated how to use the matplotlib library to create simple line, bar, and pie charts from data stored in a DataFrame. Students completed short in-class activities that mirrored the demonstrations to develop their competency in accomplishing these tasks.

A Taste of Data Analytics: Twitter Analytics
The remainder of the second and third sessions introduced an application for analyzing Twitter data using pandas. The instructors shared an example based on Mayo (Mayo, 2017) showing how to read and analyze a file of Tweets obtained from a user's Twitter account. Determining the total number of Tweets, most popular retweets and likes is accomplished by storing the tweets in a DataFrame, and sorting or summarizing the appropriate columns using methods of that class.

The corresponding homework assignment had students use the pandas, numpy, and matplotlib modules to analyze popular hashtags or mentions from a file of Tweets. A hashtag is any word in a Tweet that begins with a # symbol. A mention is any word in a Tweet that begins with an @ symbol. Students could analyze a file provided by the instructors containing Tweets from a university Twitter account or create a file containing their own Tweets. The assignment required students to import the Tweets into a DataFrame, process the text of each Tweet to create a dictionary of hashtags or mentions and their frequency of use, sort the results alphabetically and by frequency, and plot the results on a horizontal bar chart. The complete assignment description and sample output are shown in Appendix 2.

4. METHODS AND RESULTS

This study takes a quantitative approach using a survey instrument. In addition, student comments and reflections provide qualitative examples to support the results and conclusions.

Seventy of 75 students enrolled in the three spring 2018 sections voluntarily completed an anonymous online survey at the start and end of the semester to share their learning, interest, and impressions of the Python course. The incentive for completing the survey was three extra credit points added to their final exam scores. The discussion that follows presents results from that survey.

The classes were gender balanced, with 51% males, and 49% females. Fifty-four percent of the students were seniors, 36% juniors, 8% sophomores, and 1% freshmen. Of the students enrolled, 48% had taken or were currently taking a first programming course in Java, 15% had taken or were currently taking a second Java programming course, and 37% had taken or were currently taking an introductory web development course covering HTML and JavaScript. Seventeen students claimed to have taken a programming class in high school. Appendix 1, Table 3 shows the variety of students' majors and minors enrolled in the course. 43% of the students registered were CIS majors and 35% were CIS minors. Of the non-CIS majors, 50% were other non-CIS business majors and 7% were arts and science majors across 17 undergraduate programs of study.

As a response to the first research question, "What factors motivate students to take (and universities to offer) Python as part of the business curriculum?" students cited a desire to increase their future career opportunities and interest in programming as the two most common reasons for taking this class. Sample responses included "I enjoy the challenge of programming" and "My internship for this coming summer asked me if I could get into a Python programming class to be more helpful on the trading floor."

Students with little or no prior coding experience found the course to be extremely difficult. Short group projects assigned during class time created an active learning experience which enabled those with prior experience to assist their classmates for whom programming was new. Students recognized the different skill levels of their classmates and suggested that the CIS department consider offering both introductory and intermediate level Python courses in the future.

Of the assignments described in Appendix 1, Table 2, students found the first two assignments (About Me and Calculations) to be relatively easy, as expected. These programs required writing simple print statements and performing sequential calculations. The Buzz simulation described in (Offenholley, 2012) required converting values between numbers and string representations, conditional statements and loops as well as formatting data using format strings. This assignment proved to be more challenging as students were not used to integer division or modulus operations. The Account and Donor Management projects were even more challenging because of their complexity and the need for conceptual familiarity required when writing code to iterate through collections and files. The Battleship game proved to be very difficult because it required both an understanding of objects and classes, and the ability to develop relatively complex logic to place ships randomly in a grid. Students found the Data Analytics assignment at the end of the semester to be a welcome change because of its inherent relevance and much more manageable scope. The assignment required writing far fewer lines of code than the three previous assignments, and its solution closely mirrored examples shown in class.

Usefulness and Relevance of Course Topics

Students reflected on the course topics as related to their usefulness in understanding programming, as well as on their relevance to future careers. Appendix 1 Figures 1 (a) and (b) summarize these results. It is interesting to note that students found the data analytics module to be the most relevant while they considered all the other topics more useful in contributing to their overall knowledge of programming. This may be because creating an application using pandas and related libraries does not develop new programming skills, but rather, provides a relevant context in which to apply Python skills developed earlier in the semester. While mastering programming concepts provide the foundation for building Python applications, students found building applications to be more pertinent to their future careers.

In response to the second research question regarding which topics and applications of Python are considered relevant in a business context, students mentioned data analytics, dictionaries, lists and functions most frequently. One student said, "Pandas and matplotlib were cool, I wish we could have done less restrictive projects using those libraries.

Python and Employability

To answer the third research question on the value that business students receive from taking an introductory Python course, students reflected in the end-of-semester survey on the importance of knowing how to write code even if it is not a job requirement, the ability to write code as it relates to their commitment to become IT professionals, the importance of knowing Python when applying for future jobs or internships, the extent to which employers value employees who have Python skills, the extent to which of having Python skills increases their value in the job market, compared to students without Python skills, and their abilities to tackle real-world problems and projects in their future work. Most students somewhat or strongly agreed with each of these and related statements, as shown in Appendix 1, Figure 2.

Business students in majors other than CIS found value in the course as they will apply their knowledge to in their future careers. A data analytics major wrote, "Python will be very relevant in my future career." Another student said, "Midway through the course I looked up how to apply analytics to Python and saw the pandas module. I didn't know how to download it or go about it, so I waited to see if we would get to it at the end of the semester. We ended up getting to it but had 2 classes on it. If we went more in depth with Pandas I think knowing that could help me more in my job as an investment analytics associate at a media agency this August." Another student commented, "I am a marketing major. Often, I need to understand the context of discussions around a product/topic. Python can [be used to] design web-crawlers and extract those data automatically. With this technology, I do not have to passively observe discussions anymore."

6. CONCLUSIONS AND FUTURE WORK

Teaching an introductory Python course with no prerequisites to a classroom of students from multiple business majors, each with varying experiences in developing code, was a challenge. Grouping students with prior programming knowledge with their less-experienced classmates was one way to bridge the gap between students majoring or minoring in CIS and other business disciplines. By completing several smaller in-class programming problems, students were prepared to work on larger homework projects.

Replacing graphics processing and user interface development with a module on pandas and data analytics was a favorable change in the course during the spring 2018 semester. Using these tools to analyze Twitter data has more immediacy and relevance than a more abstract or contrived textbook example. Based on student evaluations after teaching the course, students would prefer additional exposure to data analytics topics. In future semesters, the course will increase coverage of data analytics topics, replacing advanced topics such as recursion, and polymorphism.

As business curricula increase their major and minor offerings to include programs of study related to data analytics, such as Data Analytics, Finance and Technology, and Auditing Analytics), the demand for Python instruction will continue to increase. Introducing data analytics-related examples to the introductory Python course will make its content more relevant and offer wider appeal to the variety of students enrolled.

7. ACKNOWLEDGEMENTS

The authors acknowledge Professor Wendy Lucas for proposing this course at Bentley University, and for shepherding it through the curriculum committee for approval, and student tutors for their careful review of the assignments.

8. REFERENCES

Cass, S. (2017, July 18). The 2017 Top Programming Languages. *IEEE Spectrum: Technology, Engineering, and Science News*. Retrieved May 23, 2018, from https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages

Continuum Analytics. (2018). Miniconda — Conda. *Miniconda - Conda*. Retrieved May 24, 2018, from https://conda.io/miniconda.html

Downey, A. (2016). Think Python, 2nd Edition. *Green Tea Press*. Retrieved June 4, 2018, from http://greenteapress.com/thinkpython2/html/index.html

Frydenberg, M. (2017). Ding Dong, You've Got Mail! A Lab Activity for Teaching the Internet of Things. *Information Systems Education Journal*, *15*(2), 20.

Guo, P. (2014, July 7). Python Is Now the Most Popular Introductory Teaching Language at Top U.s. Universities. Retrieved May 23, 2018, from https://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-u-s-universities/fulltext

Kortsarts, Y., Morris, R., & Utell, Janine. (2010). Interdisciplinary Introductory Course in Bioinformatics. *Information Systems Education Journal*, *8*(27). Retrieved June 3, 2018, from http://isedj.org/8/27/index.html

Lambert, K. (2018). *Fundamentals of Python: First Programs* (2nd ed.). Cengage. Retrieved June 4, 2018, from https://www.cengage.com/c/fundamentals-of-python-first-programs-2e-lambert/9781337560092

Mayo, M. (2017, March). A Beginner's Guide to Tweet Analytics with Pandas. Retrieved May 23, 2018, from https://www.kdnuggets.com/2017/03/beginners-guide-tweet-analytics-pandas.html

McMaster, K., Sambasivam, S., Rague, B., & Wolthuis, S. (2017). Java vs. Python Coverage of Introductory Programming Concepts: A Textbook Analysis. Information Systems Education Journal, 15(3), 4.

Offenholley, K. (2012). Gaming Your Mathematics Course: The Theory and Practice of Games for Learning. Journal of Humanistic Mathematics, 2(2), 79–92.

Reynolds, J. (2018, February 8). 8 World-Class Software Companies That Use Python. Real Python. Retrieved June 1, 2018, from https://realpython.com/world-class-companies-using-python/

Richardson, L. (2017, August 11). Beautiful Soup: We called him Tortoise because he taught us. Retrieved June 4, 2018, from https://www.crummy.com/software/BeautifulSoup/?

Siegfried, R. M., Siegfried, J., & Alexandro, G. (2016). A Longitudinal Analysis of the Reid List of First Programming Languages. Information Systems Education Journal, 14(6), 47.

Topi, H., Valacich, J. S., Wright, R., Kaiser, K., Nunamaker, J. F., Sipior, J., & de Vreede, G. J. (2010). IS 2010 Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. Association for Computing Machinery. Retrieved June 4, 2018, from https://www.acm.org/binaries/content/assets/education/curricula-recommendations/is-2010-acm-final.pdf

US News & World Report. (2018). The 10 Best Colleges for Management Information Systems. Retrieved June 3, 2018, from https://www.usnews.com/best-colleges/rankings/business-management-information-systems

Appendix 1. Additional Tables and Figures

Table 1. Python at Top 12 MIS Programs (according to US News)

Rank	Name	Python is required as a Core Course for IS Majors	Python is offered an elective for IS Majors
1	Massachusetts Institute of Technology	Python is one of eight required courses for business analytics major (MIT has no IS undergraduate program)	No
2	Carnegie Mellon University	No	Python is only offered at graduate level (MSBA)
3	University of Arizona	No	No
4	University of Minnesota Twin Cities	2 object-oriented courses (can be Python or other OO language)	No
5	University of Texas Austin	1 app development course	No
6	Georgia Institute of Technology	Java required, not Python	No
7	Indiana University Bloomington	No	Yes
8	University of Maryland College Park	No	No
9	University of Pennsylvania	No	No
10	Georgia State University	1 object-oriented course (can be Python or other OO language)	No
11	University of Michigan Ann Arbor	N/A	N/A
12	New York University	1 object-oriented course (can be Python or other OO language)	No

Table 2. Homework Assignments (Spring, 2018)

#	Topics	Description
1	Displaying information, using an IDE	Print information about you
2	Expressions and Data Types	Calculate unit prices of food items based on quantity purchased
3	Control Structures	Buzz Game (test for numbers containing or divisible by 7) (Offenholley, 2012)
4	Strings, Text Files, Lists, OS Module	User Account Manager – store usernames, passwords, allow users to add/edit/delete account information
5	Dictionary of Lists	Manage a list of donors and donation amounts; determine most generous donor, and total donations.
6	Classes and Objects	Hide ships on a Battleship game grid; a player must locate them all within a specified number of turns
7	Introduction to Data Analytics	Analyze a file of Tweets to determine most popular hashtags; create a horizontal bar chart showing hashtags and frequency

Table 3. Majors and Minors enrolled in 3 sections.

The number of students from each major is shown in bold. Minors of students with identified majors shown are indented.

Accountancy	**1**
None Specified	1
Actuarial Science	**7**
Business Studies	4
Computer Information Systems	1
Data Technologies	1
English	1
Business Studies	**3**
Computer Information Systems	2
None Specified	1
Computer Information Systems	**31**
Data Technologies	1
Finance	4
Information and Process Management	2
Information Design and Corporate Communication	1
Leadership	1
Management	2
Marketing	1
Mathematical Sciences	1
Natural Sciences	1
Philosophy	1
Psychology	1
None Specified	15
Corporate Finance and Accounting	**1**
Computer Information Systems	1
Data Analytics	**1**
Business Studies	1
Economics-Finance	**4**
Computer Information Systems	3
None Specified	1
Finance	**8**
Accountancy	1
Computer Information Systems	6
Mathematical Sciences	1
Information Design and Corporate Communication	**1**
Computer Information Systems	1
Information Systems Audit and Control	**2**
Computer Information Systems	2
Liberal Studies	**1**
Computer Information Systems	1
Management	**2**
Computer Information Systems	1
Mathematical Sciences	1
Managerial Economics	**1**
Computer Information Systems	1
Marketing	**5**
Computer Information Systems	4
None Specified	1
Mathematical Sciences	**2**
Computer Information Systems	1
Sociology	1
Public Policy	**1**
Computer Information Systems	1
None Specified	**1**
None Specified	1
Grand Total	**72**

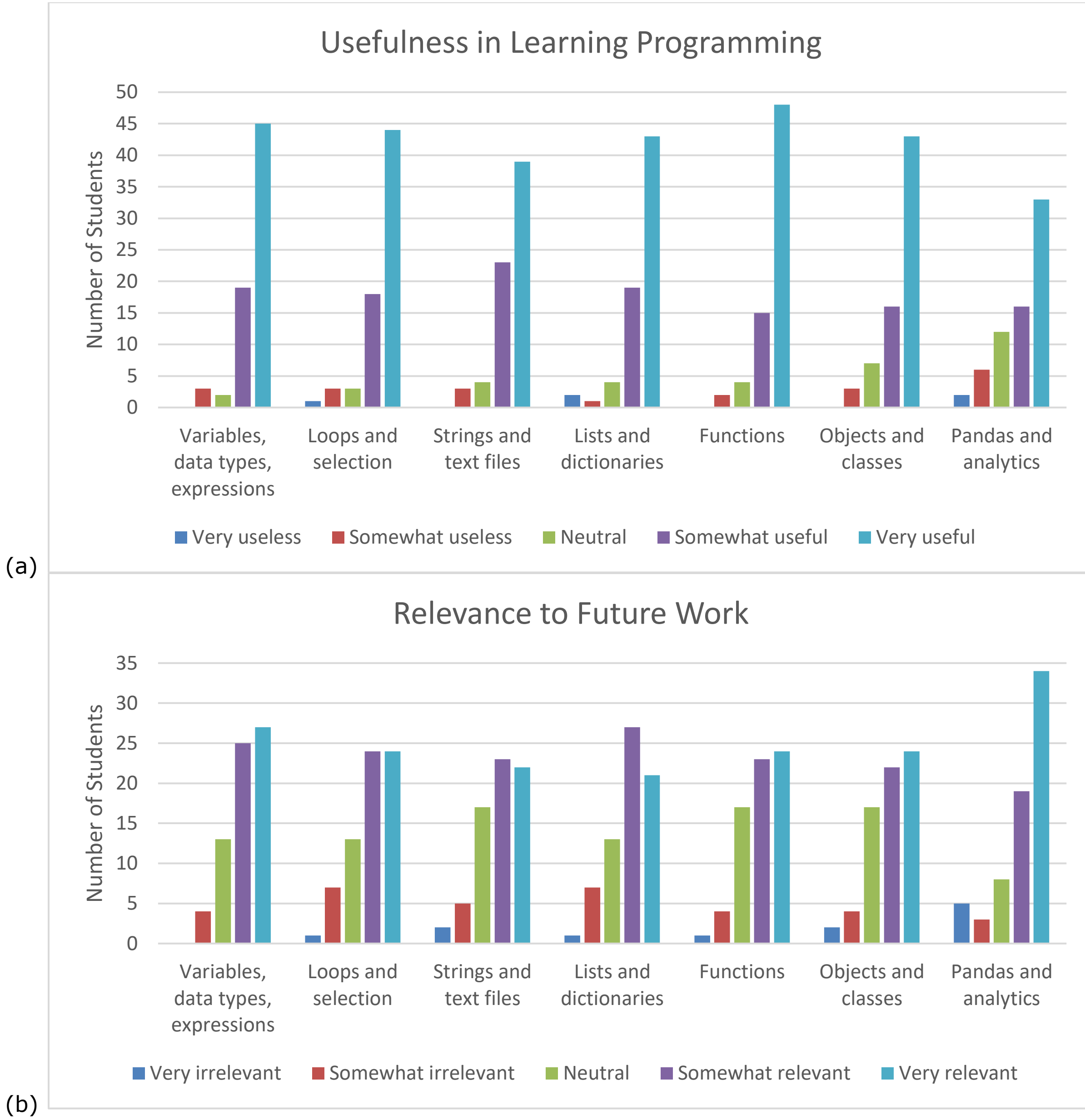


Figure 1. Python Topics: (a) Usefulness in Learning Programming and (b) Relevance to Future Work.

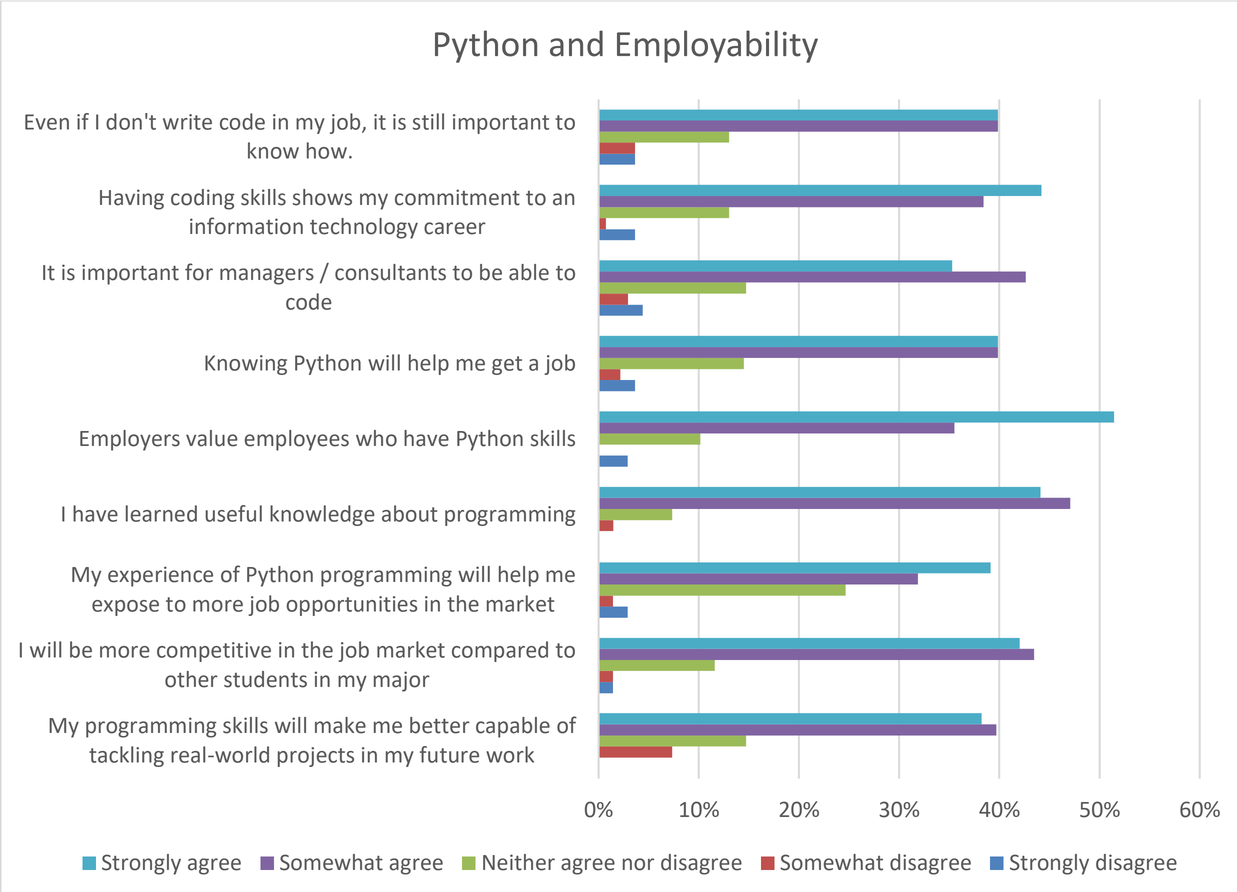
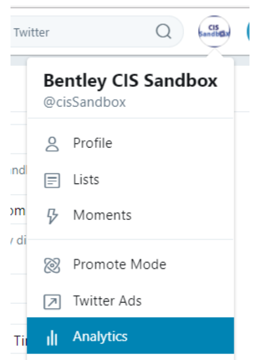


Figure 2. Perceived Value of Python Skills as related to Employability

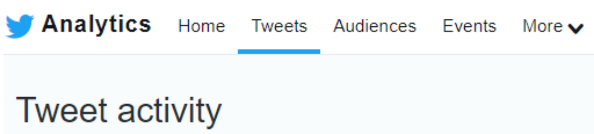
Appendix 2. Twitter Analytics Assignment Description

A **hashtag** is any word in a Tweet that begins with a # symbol, a **mention** is any word in a Tweet that begins with an @ symbol. In this assignment, you will use the pandas, numpy, and matplotlib modules to analyze hashtags or mentions found in Tweets during a period of several months and exported from Twitter.

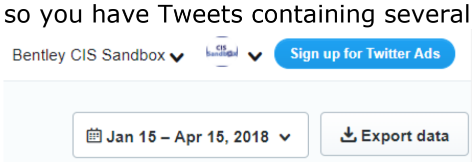
Export Your Tweets
You can use the sample Twitter data file supplied containing Tweets from a university Twitter account or export your own Tweets to analyze. Follow these steps to export your own Tweets:



1. Sign in to your Twitter account.
2. Click **your profile icon** at the top right corner of the page.
3. Click **Analytics** from the dropdown menu below your profile icon.
4. On the Analytics page, click **Tweets** at the top of the page



5. Select a date range of activity (choose a large enough date range so you have Tweets containing several hashtags or mentions),



6. Click the **Export Data** button to export your Twitter data.
7. Save the file as tweets.csv in the folder with the python program you are about to write.

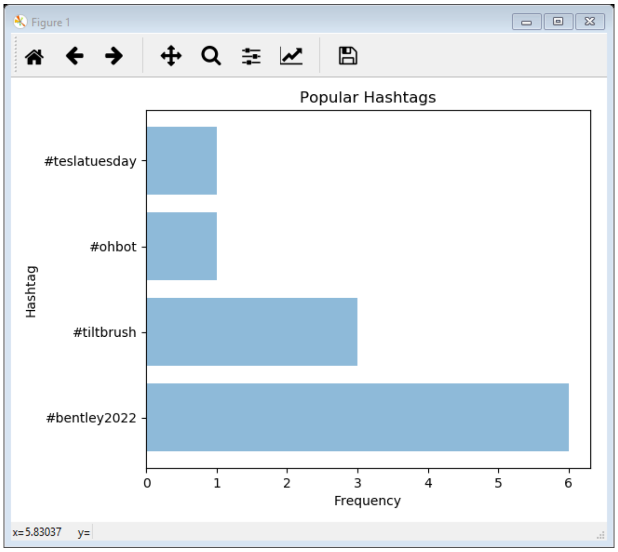
Analyze Your Tweets

1. Follow the example given in class to load Twitter data from the tweets.csv file into a pandas DataFrame, removing irrelevant columns.
2. Allow the user to specify whether to analyze hashtags or mentions by typing h or m.
3. Analyze the **Tweet text** column of the data frame to create a dictionary of hashtags/mentions and corresponding frequencies (number of times each appears). For simplicity, the DataFrame will have two columns: Hashtag and Frequency, regardless of whether you are analyzing hashtags or mentions. As you process the data, convert all words to lower case so that, for example, hashtags #PYTHON and #python would be considered the same.
4. Create a pandas DataFrame containing each hashtag and its corresponding frequency.
5. Print the DataFrame.
6. Sort the DataFrame alphabetically by hashtag. Print the sorted DataFrame.
7. Sort the DataFrame in decreasing order by frequency. Print the sorted DataFrame.
8. Create a horizontal bar chart that plots each hashtag and its frequency. Be sure to set x and y labels for the axes, ticks for values and hashtags, and a title for your plot.

See https://matplotlib.org/gallery/lines_bars_and_markers/barh.html or https://medium.com/python-pandemonium/data-visualization-in-python-bar-graph-in-matplotlib-f1738602e9c4 for examples of how to create a horizontal bar chart.

Sample Output

```
Analyze [H]ashtags or [M]entions? h
         Hashtag  Frequency
0   #bentley2022          6
1         #ohbot          1
2  #teslatuesday          1
3     #tiltbrush          3
Sorted Alphabetically
         Hashtag  Frequency
0   #bentley2022          6
1         #ohbot          1
2  #teslatuesday          1
3     #tiltbrush          3
Sorted Descending by Frequency
         Hashtag  Frequency
0   #bentley2022          6
3     #tiltbrush          3
1         #ohbot          1
2  #teslatuesday          1
```



A Sentiment and Linguistic Analysis of Online Reviews of Online Universities

Alan Peslak
Arp14@psu.edu
Information Sciences and Technology
Penn State University
Dunmore, PA 18512

Abstract
The growth and proliferation of online higher education have been staggering. This comes despite an overall decline in university enrollments. But the quality of online courses has been questioned by many researchers, suggesting it may be less than a traditional face to face experience. Many researchers have explored this area but our research study reviews online universities from an impartial non-profit organization and analyzes both the content and the sentiment in these reviews. This analysis provides a unique perspective on how online universities and their programs are viewed by their students. Implications of this analysis are examined.

Keywords: online education, reviews, sentiment analysis, linguistic analysis, online universities, online reviews

1. INTRODUCTION

The growth of online studies in higher education has been phenomenal. The number of students taking at least one online university course has grown from 1.2 million in 2002 to 5.8 million in 2014 (WICHE, 2016). According to the 2016 Online Report Card that Babson Survey Research Group publishes yearly, nearly 1/3 (32%) of all higher education students are enrolled in at least one online course. Of this 32%, 15% are fully online and 17% take a mix of traditional delivery with some online courses. (Seaman, Allen and Seaman, 2016). The concentration of distance education enrollments is noteworthy. "Almost half of distance education students are concentrated in just five percent of institutions, while the top 47 institutions (just 1.0% of the total) enroll 22.4% (1,421,703) of all distance students." (Seaman et al. 2016). The growth in online education has been non-stop with increased enrollments for the fourteenth straight year. This growth in online students came to a large extent at the expense of face-to-face instruction. Overall, students in higher education declined by a total of 800,000 students between 2012 and 2016. This means that with the growth in online education total face-to-face students are down more than one million students over the past 4 years. (Seaman, Allen, and Seaman, 2016). Given this tremendous growth in online higher education, it is essential that the quality of online higher education be thoroughly examined. Many studies have been prepared that begin to address this issue. Specifically, our research questions are what are the current views of students of online higher education. We see student opinions as an approach to understanding and estimating the current quality of online higher education. Our method is to analyze the current status via a review of social media, specifically by analyzing sentiment and linguistics of online reviews of online higher education institutions. These reviews are the opinions of students who have taken courses in these universities. They will only serve as a surrogate to overall quality but student opinions and reviews of course quality have shown to correlate strongly with other means of evaluations (Marsh, 1987).

2. LITERATURE REVIEW

Many researchers have explored the attitudes of students regarding the quality and effectiveness of their online university classes. Tallent-Runnels (2006) reviewed all research literature in 2006 and found many problems and inconsistencies in online learning. Braun (2008) performed a much-cited study of 90 grad students at one Western US university. He found that 90% of students found online courses were viewed as more challenging than traditional face to face courses and that only 8% would not want to take an online course again. Margaryan, A., Bianco, M., & Littlejohn, A. (2015) found that many Massive Open Online Courses (MOOCs) had poor instructional design.

There is also research that suggests that online education has improved over time. Allen and Seaman (2016) found 57% of students rated their online graduate courses as good as or better than face to face in 2003. This percentage increased over the years with 77% of respondents rating online as good or better. This is strong improvement over the decade and comes along with the growth and proliferation of online higher education.

Our research analyzes online reviews of online universities. The study of user reviews has a robust track in current research. Much work has been accomplished in analyzing product reviews on sites such as Amazon (Mudambi, & Schuff, 2010) Dellarocas, Zhang, & Awad (2007) and Duan, Gu, & Whinston (2008) analyzed movie sales and popularity based on online movie reviews and found a positive correlation. Ye, Law, & Gu (2009) studied the impact of online reviews and their effect on hotel room bookings. Likewise, Clemons, Gao, & Hitt (2006) studied how online reviews affect craft beer differentiation. Hu, Bose, Koh, & Liu (2012) began work on analyzing the sentiment of online reviews. Their work centered on use of sentiment analysis to determine bias and manipulation of reviews. Bowen, Chingos, Lack, & Nygren (2014) concluded that if online courses are well designed that "have the potential" to achieve "equivalent educational outcomes"

There has also been preliminary work done on what has come to be known as E-WOW or electronic word of mouth. Filieri, & McLea (2014) analyzed social media and reviews of travel sites to determine relevant variables.

As to validity of ratings, Marsh (1987) generally found that student ratings of courses generally were "valid against a variety of indicators of effective teaching". Therefore, we believe we can use online students reviews as a relatively accurate reflection of learning effectiveness.

3. METHODOLOGY

Based on past studies, we have selected student online opinions as the basis for determining perceived relative views of online universities. These student opinions will then serve as a proxy for perceived quality of the online university program. We thus searched for a reliable review source available for study.

All reviews have been obtained from the website Guide to Online Schools. Data were gathered in February of 2018. https://www.guidetoonlineschools.com/online-reviews Permission has been obtained to use this data. The parent organization which solicits and hosts these reviews is the Washington based SR Education Group. Their overall mission is

> "Our products are designed to help prospective students find a college suited to their individual needs, whether that means low tuition costs, high satisfaction reported by recent graduates, or degrees that lead to career advancement. We feature schools with great student reviews and strong success metrics, and provide unbiased, comprehensive information."

Each review was analyzed for content and sentiment.

Sentiment analysis (Polarity) was obtained via Meaning Cloud for Excel and summary results for each were analyzed via SPSS (N=negative, P=Positive, etc.)

The following glossary from Meaning Cloud (2018)describes their coding:

"Agreement This field marks the agreement between the sentiments detected in the text, the sentence or the segment it refers to. It has two possible values:

- •AGREEMENT: the different elements have the same polarity.
- •DISAGREEMENT: there is disagreement between the different elements' polarity.

Subjectivity This field marks the subjectivity of the text. It has two possible values:

- •OBJECTIVE: the text does not have any subjectivity marks.
- •SUBJECTIVE: the text has subjective marks.

Confidence This field represents the confidence associated with the sentiment analysis performed on the text. Its value is an integer number in the 0-100 range.

Irony This field indicates the irony of the text. It has two possible values:

- NONIRONIC: the text does not have ironic marks
- IRONIC: the text has ironic marks

Detection of irony
identifies comments in which what is expressed is the opposite of what is said.

Graduated polarity
distinguishes very positive and very negative opinions, as well as the absence of sentiment.
Agreement and disagreement
identifies opposing opinions and contradictory or ambiguous messages."

In addition, LIWC (Linguistic Analysis and Word Count) from Pennebaker was used to analyze linguistical meaning embedded in the reviews. The use of LIWC is well established in the literature. Robinson, Navea, and Ickes (2013) used LIWC analysis of students written self-introductions to grades that students achieved. Cordova, Cunningham, Carlson, and Andrkowski (2001) used LIWC to analyze how individuals adjusted to having breast cancer. There are many more examples of the use of LIWC used for scholarly research.

LIWC (Linguistic and Word Count) software (Pennebaker, Booth, Boyd, and Francis, 2015) is one of the most accepted and popular linguistic analysis tool. "The way that the **L**inguistic **I**nquiry and **W**ord **C**ount (LIWC) program works is fairly simple. Basically, it reads a given text and counts the percentage of words that reflect different emotions, thinking styles, social concerns, and even parts of speech. Because LIWC was developed by researchers with interests in social, clinical, health, and cognitive psychology, the language categories were created to capture people's social and psychological states. …. The text analysis module then compares each word in the text against a user-defined dictionary. As described below, the dictionary identifies which words are associated with which psychologically-relevant categories.

After the processing module has read and accounted for all words in a given text, it calculates the percentage of total words that match each of the dictionary categories. For example, if LIWC analyzed a single speech that was 2,000 words and compared them to the built-in LIWC2015 dictionary, it might find that there were 150 pronouns and 84 positive emotion words used. It would convert these numbers to percentages, 7.5% pronouns and 4.2% positive emotion words." (Pennebaker Conglomerates, 2015).

LIWC was used in our study to enhance understanding of online review content as well as to enhance our findings.

4. RESULTS

The results of our sentiment and linguistic analyses are presented in the following tables.

		Frequency	Percent
Valid	N+	9	.7
	N	176	13.6
	NEU	141	10.9
	NONE	6	.5
	P	835	64.5
	P+	127	9.8
	Total	1294	100.0

Table 1 Meaning Cloud Polarity

Overall results of all reviews and the Polarity of each review are shown in table 1. The scale runs from N+ (very negative), N (Negative), NEU(Neutral), None, P (Positive) and P+ (Very Positive) and reflects the overall sentiment of the review. Specific dictionaries and other analytical techniques are used by Meaning Cloud to determine whether a particular review expresses an overall good (positive) or bad (negative) expression. According to Meaning Cloud "Our Sentiment Analysis API performs a detailed, multilingual sentiment analysis on information from different sources."

The text provided is analyzed to determine if it expresses a positive, neutral or negative sentiment (or if it is impossible to detect). In order to do so, the individual phrases are identified and the relationship between them is evaluated, which results in a global polarity value of the text as a whole.

In addition to the local and global polarity, the API uses advanced natural language processing techniques to detect the polarity associated with both the entities and the concepts of the text. It also allows users to detect the polarity of entities and concepts they define themselves, which makes this tool applicable to any kind of scenario."

The table 1 results indicate that 74% of the posted reviews were positive or very positive. In addition, 11% were neutral. The overall sentiment analysis of the 1295 results (6 showed no sentiment) clearly suggests that online universities are currently viewed favorably by an overwhelming majority. Table 2 however, does suggest that this may not be unqualified. The reviews often have mixed emotions with 800 showing some disagreement, whereas only 494 showed agreement. Different elements of the texts have different sentiment or polarity.

		Frequency	Percent
Valid	AGREEMENT	494	38.2
	DISAGREEMENT	800	61.8
	Total	1294	100.0

Table 2 Meaning Cloud Agreement/Disagreement

Irony was not apparent in the online university reviews with only 30 of the 1294 reviews expressing ironic marks as shown in Table 3.

		Frequency	Percent
Valid	IRONIC	30	2.3
	NONIRONIC	1264	97.7
	Total	1294	100.0

Table 3 Meaning Cloud Irony Measure

As expected with reviews, the vast majority of the reviews (85%) are of a subjective nature (Table 4), expressing individual opinions rather than objective facts. Chi Square analysis confirms the significance of this finding at $p < .001$. (Table 5 and 6)

		Frequency	Percent
	OBJECTIVE	204	15.8
	SUBJECTIVE	1090	84.2
	Total	1294	100.0

Table 4 Meaning Cloud Subjectivity

	OBJECTIVE	SUBJECTIVE	Total
Count No Resp.	7	0	0
Expected Count	.0	1.1	5.9
% within Agreement	100.0%	0.0%	0.0%
Count	0	110	384
Expected Count	2.7	77.5	413.9
AGREE % within Agreement	0.0%	22.3%	77.7%
Count	0	94	706
Expected Count	4.3	125.4	670.3
DIS % within Agreement	0.0%	11.8%	88.3%
Total Count	7	204	1090
Expected Count	7.0	204.0	1090.0
% within Agreement	0.5%	15.7%	83.8%

Table 5 Subjectivity/Expected Levels

	Asymptotic Significance (2-sided)
Pearson Chi-Square	.000
Likelihood Ratio	.000
N of Valid Cases	

Table 6 Chi Square Significance Variation

The next analysis performed was to determine whether all online universities were viewed the same or whether there were differences in polarity based on the university. A summary by University was performed and via SPSS ANOVA. We found that mean scores were significantly different among the 340 universities at $p < .003$.

There is significant difference between universities with regard to polarity.
Overall averages by school show that there were 341 schools represented and the overall mean was 3.8, nearly a full 4.0 positive (table 8). No schools had an average score of N+ or very negative.

	N	Min	Max	Mean
Mean	341	2.00	5.00	3.8143
Valid N (listwise)	341			

Table 8 Central Tendency of Polarity

A word count analysis was also performed and relevant value-laden and associated words were collected and shown in table 9. As is apparent, most words are positive with great, well, good, helpful and best all with over 150 mentions in the content. This supports the preliminary findings of generally positive emotions associated with online universities today.

Order	Unfiltered word count	Occurrences	%
56	great	474	0.273
88	well	280	0.161
92	recommend	269	0.155
98	good	250	0.144
111	helpful	210	0.121
149	best	159	0.092
153	however	151	0.087
166	easy	139	0.08
171	nothing	136	0.078
172	better	136	0.078
176	support	134	0.077
187	care	126	0.073
195	hard	121	0.07
203	different	113	0.065
205	challenging	111	0.064

Table 9 Value Laden Words and Count

In general, online university programs are viewed favorably with 74% of the 1294 reviews from 340 online universities.

Linguistic Results

Linguistic analysis was performed using LIWC software and is shown in table 10.

The linguistic analysis yielded some interesting results about the type and style of reviews for online universities.

According to Pennebaker, Booth, Boyd, and Francis (2015) "Analytical thinking -- a high number reflects formal, logical, and hierarchical thinking; lower numbers reflect more informal, personal, here-and-now, and narrative thinking." The analytic measure had a mean for all 1294 reviews of 61.4908 which suggests a mid-high range of logical thought versus informal thought. Standards for different types of communication are shown in table 11 for comparison. This table shows scores for each analytic measure for common forms of communication such as NY Times, expressive writing, etc. As the table shows, blog posts have a much lower rating than these reviews, at 49.89 and natural speech is only 18.43. This suggests that significant thought was incorporated into the review and their reliability can be considered high. These do not appear to be random thoughts, but well thought-out opinions.

"Emotional tone -- a high number is associated with a more positive, upbeat style; a low number reveals greater anxiety, sadness, or hostility. A number around 50 suggests either a lack of emotionality or different levels of ambivalence." (Pennebaker, Booth, Boyd, and Francis, 2015). The mean tone of the reviews also was fairly high at 67.3723. This supports the positive polarity found via Meaning Cloud since a higher number reflects an upbeat versus hostile tone. Surprisingly, the tone is about equal to the New York Times at 68.17 and much more upbeat than blogs at 54.5.

"Clout -- a high number suggests that the author is speaking from the perspective of high expertise and is confident; low Clout numbers suggest a more tentative, humble, even anxious style." (Pennebaker, Booth, Boyd, and Francis, 2015). The mean clout variable expresses the degree of confidence expressed in the review. This was fairly low at 44.6430 and about on par with blogs. We see this as confirming the Meaning Cloud finding of a high level of Disagreement in the reviews. Reviewers seem to have presented both sides of the issue in their reviews, both positive and some negative and this reflects in a humbler clout factor.

"Authentic -- higher numbers are associated with a more honest, personal, and disclosing text; lower numbers suggest a more guarded, distanced form of discourse". (Pennebaker, Booth, Boyd, and Francis, 2015). The authenticity or honest communication aspect of the reviews shows a mean of 50.3574, nearly exactly in the middle of the honest to guarded spectrum. Whereas the NY Times has a rating of high honesty at 24.84 since they present mostly facts, the reviews reflect an even mix between honest facts and opinion.

	Mean	Std. Deviation
Analytic	61.4908	24.20618
Tone	67.3723	32.18595
Clout	44.643	25.16279
Authentic	50.3574	29.64372
WPS	17.3064	7.37118
Sixltr	22.7386	7.18486
Dic	89.4139	7.84134
Function	52.8078	7.19261
Pronoun	13.6895	4.88103

Table 10 LIWC results of Reviews

Category	Blogs	Expressive Writing	Natural Speech	NY Times	Twitter
Analytic	49.9	44.9	18.4	92.6	61.9
Clout	47.9	37.0	56.3	68.2	63.0
Authentic	60.9	76.0	61.3	24.8	50.4
Tone	54.5	38.6	79.3	43.6	72.2
Words/sentence *	18.4	18.4	-	21.9	12.1
Words> 6 letters	14.4	13.6	10.4	23.6	15.3
Dictionary words	85.8	91.9	91.6	74.6	82.6
Total function words	53.1	58.3	56.9	42.4	46.1
Total pronouns	16.2	18.0	20.9	7.4	13.6

Table 11 LIWC standards (Pennebaker, 2015)

Further analysis of the reviews for online universities was performed with LIWC and included WPS or words per sentence, Sixltr or number of words of six or more letters, and Dic or Dictionary words to determine complexity and level of writing. Function words are non-content words. Function words act as connectors between meaningful content words. Pronouns such as (I, me, you) suggest the written words have a more personal comment. These measures in table 10 compared to table 11 show mixed levels of complexity. WPS and Dictionary words match about at Blog level but Sixltr words are about at NY Times levels. Function words are also at blog level and pronouns match up to Twitter, well above NY Times levels but below Blogs and expressive writing. Overall, the reviews are fairly complex, fairly personal, and contain relatively high levels of non-content words.

A final correlation analysis was performed to determine what linguistic aspects of the review correlated with polarity (positive or negative evaluation).

Category	Correlation	Sig. (2-tailed)
Polar	1	
WC	-.154**	0
Analytic	.220**	0
Clout	-0.004	0.899
Authentic	-0.054	0.05
Tone	.622**	0
Sixltr	.232**	0
Dic	-0.049	0.078
Function	-.202**	0
Pronoun	-.149**	0

Table 12 LIWC and Polarity Correlation and Significance

The results of this analysis are shown in table 12. Many linguistic measures have significant correlation with degree of polarity. Higher word count is negatively correlated which suggests that more words in the review, the more it tends to be negative. Conversely shorter reviews tend to be positive. The analytic measure correlates positively with polarity and suggests more thoughtful and detailed reviews tend to be positive. Negative reviews provide less analytical support. Tone has high and positive correlation with polarity as anticipated since positive tone would suggest positive comments. This analytic also supports our findings from the Meaning Cloud results. Six letter or longer words positively correlate with polarity with positive reviews having more complexity. Both function words and pronouns are negatively correlated with positive polarity suggesting again that more meaningful words are included in thoughtful positive reviews. Clout and the number of dictionary words do not have significant correlation.

5. CONCLUSION

The results of the over 1290 reviews show that overall sentiment is generally positive. 74% of reviews are positive or very positive. Only 14% are negative or very negative clearly the results indicate that online universities have achieved an overall positive response from its students and other related stakeholders who have opted to complete a review. Since we are using student opinions as a proxy for quality, this suggests that overall quality of online universities is good. But not all online universities are viewed as good as others. The overall mean of our sample was 3.81 on a scale of 1 to 5. But there were statistically significant differences among the 340 universities at $p < .003$. The range was from negative to very positive. Not all schools are viewed positively. We recommend that students review ratings for the specific online university they are exploring to assure they are getting a school with positive ratings. For the schools themselves, they are recommended to review their ratings and address shortcomings that have been expressed.

Even though the reviews are generally positive, there is some level of disagreement found in the sentiment analysis. Different elements of a review have different levels of polarity. For schools this proposes that even though their review may be generally positive there still may be areas that can be improved in their online program.

The online reviews were overall found to be nonironic but also subjective. Subjectivity is expected since these are opinions students have of the program. Also, the nonironic tone portrays honest and straightforward opinions. We can thus rely on the sentiment classification.

Our review of the most common words used in reviews reinforces the findings that online universities are viewed favorably. The most six most frequent words used value laden words were positive.

Linguistic analysis also reveals strong positive tone and generally complex, authentic, and analytic levels especially in the positive reviews. This analysis serves as a check on the sentiment analysis. In all cases the findings of the linguistic analysis are consistent with the sentiment analysis.

Finally, though we believe these findings represent accurate opinions on online universities, the relationship between online ratings and actual product or service quality has not been fully studied. This is a limitation of the research. Further research needs to be performed to map online program outcomes with perceived quality opinions. This is an area that merits significant further research.

6. REFERENCES

Allen, I. E., & Seaman, J. (2016). Online Report Card: Tracking Online Education in the United States. *Babson Survey Research Group*.

Bowen, W. G., Chingos, M. M., Lack, K. A., & Nygren, T. I. (2014). Interactive learning online at public universities: Evidence from a six-campus randomized trial. *Journal of Policy Analysis and Management*, *33*(1), 94-111.

Braun, T. (2008). Making a choice: The perceptions and attitudes of online graduate students. *Journal of Technology and Teacher Education*, *16*(1), 63.

Clemons, E. K., Gao, G. G., & Hitt, L. M. (2006). When online reviews meet hyperdifferentiation: A study of the craft beer industry. *Journal of management information systems*, *23*(2), 149-171.

Cordova, M. J., Cunningham, L. L., Carlson, C. R., & Andrykowski, M. A. (2001). Social constraints, cognitive processing, and adjustment to breast cancer. *Journal of consulting and clinical psychology*, *69*(4), 706.

Cui, G., Lui, H. K., & Guo, X. (2012). The effect of online consumer reviews on new product sales. *International Journal of Electronic Commerce*, *17*(1), 39-58.

Dellarocas, C., Zhang, X. M., & Awad, N. F. (2007). Exploring the value of online product reviews in forecasting sales: The case of motion pictures. *Journal of Interactive marketing*, *21*(4), 23-45.

Duan, W., Gu, B., & Whinston, A. B. (2008). Do online reviews matter?—An empirical investigation of panel data. *Decision support systems*, *45*(4), 1007-1016.

Filieri, R., & McLeay, F. (2014). E-WOM and accommodation: An analysis of the factors that influence travelers' adoption of information from online reviews. *Journal of Travel Research*, *53*(1), 44-57.

Hu, N., Bose, I., Koh, N. S., & Liu, L. (2012). Manipulation of online reviews: An analysis of ratings, readability, and sentiments. *Decision Support Systems*, *52*(3), 674-684.

King, Robert Allen, Pradeep Racherla, and Victoria D. Bush. "What we know and don't know about online word-of-mouth: A review and synthesis of the literature." *Journal of Interactive Marketing* 28.3 (2014): 167-183.

Margaryan, A., Bianco, M., & Littlejohn, A. (2015). Instructional quality of massive open online courses (MOOCs). *Computers & Education*, *80*, 77-83.

Marsh, H. W. (1987). Students' evaluations of university teaching: Research findings, methodological issues, and directions for future research. *International journal of educational research*, *11*(3), 253-388.

Meaning Cloud Sentiment Analysis (2018) https://www.meaningcloud.com/products/sentiment-analysis

Mudambi, S. M., & Schuff, D. (2010). Research note: What makes a helpful online review? A study of customer reviews on Amazon. com. *MIS quarterly*, 185-200.

Mudambi, S. M., & Schuff, D. (2010). Research note: What makes a helpful online review? A study of customer reviews on Amazon. com. *MIS quarterly*, 185-200.

Pennebaker Conglomerates (2015) *LIWC How It Works*. http://liwc.wpengine.com/how-it-works/

Pennebaker, J.W., Booth, R.J., Boyd, R.L., & Francis, M.E. (2015*). Linguistic Inquiry and Word Count: LIWC2015*. Austin, TX : Pennebaker Conglomerates (2015). www.LIWC.net

Pennebaker, J., Boyd, R., Jordan, K., and Blackburn, K. (2015). *The development and psychometric properties of LIWC2015*. Austin, TX. University of Texas at Austin.

Robinson, R. L., Navea, R., & Ickes, W. (2013). Predicting final course performance from students' written self-introductions: A LIWC analysis. *Journal of Language and Social Psychology*, 0261927X13476869.

Seaman, I. Allen and J. Seaman (2016) Grade Increase Tracking Distance Education in The United States http://onlinelearningsurvey.com/reports/gradeincrease.pdf

SR Education Group (2018). Mission. https://www.sreducationgroup.org/#mission

Tallent-Runnels, Mary K., et al. "Teaching courses online: A review of the research." *Review of educational research*76.1 (2006): 93-135.

WICHE Cooperative for Educational Technologies(2016). WCET Distance Education Enrollment Report 2016 https://wcet.wiche.edu/sites/default/files/WCETDistanceEducationEnrollmentReport2016.pdf

Ye, Q., Law, R., & Gu, B. (2009). The impact of online user reviews on hotel room sales. *International Journal of Hospitality Management*, *28*(1), 180-182.

Storing and Querying Bitcoin Blockchain Using SQL Databases

Kwok-Bun Yue
yue@uhcl.edu

Karthika Chandrasekar
Chandrasekark@uhcl.edu

Hema Gullapalli
GullapalliH2010@uhcl.edu

Department of Computing Sciences
University of Houston-Clear Lake
Houston, TX 77058, U.S.A

Abstract

Bitcoin is the first major decentralized cryptocurrency with wide acceptance. A core technological innovation of Bitcoin is blockchain, a secure and pseudonymous general ledger that stores every Bitcoin transaction. Blockchain has received enormous attention from both the commercial and academic worlds, and it is generally recognized as the enabling technology of the Internet of Value (IoV), in which securely stored valuable entities are intended to be transferred as easily as information. Current blockchains are designed as special kinds of Online Transaction Processing (OLTP) systems, but not Online Analytical Processing (OLAP) systems. Data analytics by querying the blockchain directly can be ineffective. To incorporate the increasingly important blockchain technology into Information Systems curriculum, one approach is to store blockchain data in a SQL database, thus allowing fast data access and a simpler understanding of the underlying concepts. This paper describes our experiment of using three different methods for storing and querying Bitcoin data from SQL databases. It elaborates an assignment of querying a Bitcoin's SQL database in an undergraduate database course. The paper discusses our experience on using SQL databases for blockchain analysis, elaborates the characteristics of Bitcoin blockchain that make it an interesting database case, examines the relative merits of the three different methods, and provides suggestions on how they may be used in IS courses. Overall, we find that using SQL to query blockchains can be an effective educational technique for introducing it to IS curriculum.

Keywords: Blockchain, SQL, Bitcoin, database, query, data analytics.

1. INTRODUCTION

Bitcoin (Nakamoto, 2008) is the first major decentralized cryptocurrency with wide acceptance. It solves the double spending problem, in which a digital currency may be spent two or more times, by storing a publicly accessible general ledger of *all* Bitcoin transactions in a blockchain (Nakamoto, 2008). Unlike bank transactions, Bitcoin transactions are digitally signed and irreversible, and are stored in a peer-to-peer network of nodes (running Bitcoin Core) using the Bitcoin protocol (Antonopoulos, 2017). Bitcoin Core (Bitcoin.org, 2018) is open sourced and contains code storing and maintaining a copy of the Bitcoin blockchain in a node, together with a reference Bitcoin's client to interact with the blockchain.

Although considered as behaving more like a speculative investment than a currency by many (Yermack, 2015; Detrixhe, 2018), Bitcoin has stormed into public awareness, reaching a historical peak price of $17,900 on December 15, 2017 (Wikipedia, 2018). Table 1 contains a collection of some vital Bitcoin parameters on 1/15/2019 11:00am central time to provide an illustrative snapshot. The data is collected from various public websites, including blockchain.info, bitnodes.earn.com, and bitcoinblockhalf.com. Some parameters will be explained later in Section 2. As of January 2019, the current size of the Bitcoin blockchain is more than 199GB and it stores all of the more than 373 million of Bitcoin transactions. With an all-time high market capitalization of $281 billion reached in December 2018 seemingly pulling out of thin air, no wonder Bitcoin has caught the imagination of the public.

Number of Bitcoin nodes	10,176
Number of Bitcoins mined	17,483,325 (83.25% of total)
Bitcoin's price	$3,632
Bitcoin's market capitalization	$64,292,208,028
Bitcoin blockchain's size	199.0GB
Latest block	558,665
Number of transactions in the latest block	2,854
Estimated transaction volume in the latest block	3,530.81722584 Bitcoin (BTC)
Total transaction fees in the latest block	0.14961742 BTC
Total number of all Bitcoin transactions	373.2 millions
Difficulty level	5,883,988,430,955
Number of transactions in the last 24 hours	321,592
Number of unspent transaction outputs	60,947,620

Table 1. A snapshot of Bitcoin's Parameters on 1/15/2019 11:00am central time

Bitcoin's success triggered many other cryptocurrencies, called altcoins, which numbered in 1,565 as of April 20, 2018 (Wikipedia 2018b). Even so, many consider that the blockchain technology developed and validated by Bitcoin may be much more important than Bitcoin itself (Tucker, 2018). Bitcoin blockchain can be considered as the first generation of blockchain that stores a specific cryptocurrency. Current and future generations of blockchains advance in many directions (Zheng, et al., 2017).

With the general ledgers of transactions nearly impossible to tamper with, blockchains can be extended to store any valuable property or asset beyond cryptocurrency. An important advancement is the introduction of rich programming languages and stateful blockchains to allow the constructions of smart software contracts to govern transaction completion, such as the approach taken by Ethereum (2018), the second most popular cryptocurrency. Another key development is permissioned blockchains, such as Hyperledger's fabric (Androulaki, et al., 2018). Unlike public blockchain such as Bitcoin in which everyone can participate, permissioned blockchains allow only a set of known and identified participants, who shared a common goals but do not fully trust each other. This enable private and federated blockchains (Zheng, et al., 2017).

Tapscott, & Tapscott (2017a) indicate that blockchain technology enables businesses with the Internet of Value (IoV): "a secure platform, ledger, or database where buyers and sellers could store and exchange value without the need for traditional intermediaries." The results can be drastically reduced transaction cost and friction that disrupts the usual ways of conducting businesses in a wide spectrum of areas. Using higher education as an example, blockchain allows a Web of decentralized transactions, possibly enabling huge changes in keeping student records, optimizing student loan management, improving pedagogy, incubating meta-universities, and ultimately creating a global network of learning institutes (Tapscott & Tapscott, 2017b). However, it is worthy to note that like many other leading edge technology, blockchains come with risks and costs (for example, see Walch, 2015).

Despite its importance, information systems (IS) research in blockchain is just beginning to emerge (Beck, Avital, Rossi & Thatcher, 2017). In IS education, blockchain can be relevant to many courses, including technical topics such as computer security, data analytics, databases, cryptocurrency, smart contracts, financial technology, etc. There are very few papers on incorporating blockchain technology in information systems and computing courses, especially in the lower level. An exception is (Delmolino, et al., 2016) that describes the experience of safe smart contract development laboratories in a security class. There is a gap between the importance of blockchain, and its existing body of knowledge and results in IS education. For example, the 2017 EDSIG conference provided a workshop on "the Easy Way to Create a Blockchain using Fabric Composer" (Foley & Decker, 2017) in an effort to bridge the gap. This paper aims to contribute in filling this gap by describing our experience with storing and querying Bitcoin blockchain using SQL databases. It is possible that other popular blockchains, such as Ethereum, can be used for the same purpose of experimentation with blockchain. However, we selected Bitocin since it is the most popular public blockchain with tools widely available.

The rest of the paper is structured as follows. Section 2 discusses the basics of Bitcoin and its blockchain, and the goals of this work as the background context. Section 3 examines three methods of accessing Bitcoin data from SQL databases. Section 4 describes a Bitcoin's SQL query assignment in an undergraduate database course and the accompanying surveys. Section 5 discusses our experience using SQL to query the Bitcoin blockchain. It describes the characteristics that make it an interesting database case, and provides suggestions on how these different methods can be adopted in IS courses. Section 6 discusses future directions and draws our conclusions.

2. BACKGROUND

2.1 Bitcoin Blockchain and Transactions

Bitcoin blockchain stores the entire history of Bitcoin transactions. A transaction stores the transfers of Bitcoins (in the unit of Satoshi, with 1 Bitcoin (BTC) = 100,000,000 Satoshi) from input accounts to output accounts, plus authorization and other information. Bitcoin account addresses are public key hash values that can be authenticated by the corresponding private keys. Users can use a Bitcoin wallet to manage their Bitcoin accounts (public key hashes) and interact with the Bitcoin blockchain.

Unlike a bank transaction transferring money from one account to another account, Bitcoin transactions allow multiple inputs and multiple outputs. Figure 1 shows four historically interesting Bitcoin transactions. Figure 1a is the very first Bitcoin transaction as 50 BTC went to the Bitcoin address '1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa', which is assumed to be controlled by Satoshi Nakamoto, the mysterious Bitcoin's inventor(s). Bitcoin blockchain is known to be pseudonymous as all transactions are publicly accessible but the ownerships of accounts are anonymous within the blockchain. Many Websites provide Web pages and APIs to access Bitcoin's data in various formats such as HTML, JSON or XML. For example, one can copy and paste Bitcoin addresses, transaction hash addresses, or block addresses from this paper into the popular site, Blockchain.info. Figure 2 shows a part of the output page of Blockchain.info for the Bitcoin transaction of Figure 1a.

Transactions are grouped in blocks. For example, Table 1 indicates that the block #558,665 has 2,854 transactions. The first Bitcoin transaction, called the *genesis transaction* here, is included in the first block (known as the Genesis Block or Block #0) as shown in Figure 1a. It is known as a *Coinbase transaction* to reward 50 BTC to the Bitcoin miner who had successfully created the block. Since the reward is created out of nowhere by Bitcoin, there is no input in a Coinbase transaction. Bitcoin mining involves finding a small enough block hash of the 80 Bytes header of the new block. The required smallness, or difficulty level, of the block hash is adjusted every 2,016 blocks to ensure that every block is mined in about 10 minutes. The difficulty level of 5,883,988,430,955 in Table 1 indicates a difficulty level of more than 5 trillion times as difficult as that of the Genesis block. The 80 Byte block header contains the hash of the Merkle tree which is constructed from the hashes (addresses) of all transactions, ensuring that transactions cannot be changed. The block header also contains the previous block hash and thus the block is *chained* together. Changing a block will change its block hash, and any subsequent block hashes will needed to be recomputed. This ensures that the blockchain is nearly impossible to tamper with.

Unlike a bank that keeps the balance of every account, Bitcoin blockchain keeps track of every transaction, including those transaction outputs (TXOut) that have not yet been spent, which are known as *unspent transaction outputs* (UTXO). UTXO can be used for future transaction inputs (TXIn). Note that in Figure 2, the transaction output of the Genesis block is still an UTXO. Thus, the very first Bitcoins generated has not yet been spent, probably intentionally.

Figure 1b shows another famous transaction, a1075db55d416d3ca199f55b6084e2115b9345e 16c5cf302fc80e9d5fbf5d48d, the first documented purchase of a good with Bitcoin in which 10,000 BTC was used to buy two Domino's pizzas on May 17, 2010. This *pizza transaction* has one TXOut (presumably going to an account owned by the pizza provider). Note that the buyer gathered together 131 UTXO from previous transactions as TXIn to pool together 10,000.99 BTC. This paid the 10,000 BTC to the TXOut, and the transaction fee of 0.99BTC, which was collected by the block miner together with the 50 BTC mining reward. After the transaction was confirmed, these 131 UTXO were recorded as *spent* and can no longer be used as inputs to other transactions, thus solving the double spending problem. The current approximate 60,947,620 UTXO indicated in Table 1 is how the current Bitcoins are 'stored.'

Figure 1c shows how the 10,000 BTC were used by the 'pizza person' to provide for two TXOut in a transaction called the *pizza-provider transaction* here. Again, after this transaction, the previously unspent TXO to the pizza person with 10,000 BTC was recorded as spent.

In general, transactions can have multiple inputs and multiple outputs. Figure 1d shows the oldest Bitcoin transaction with three TXIn and two TXOut such that one TXOut address is also a TXIn address in the transaction (418b84d7649055411d8be4e241376a93825c1d 6248a304ae693060b3007a43f2). The sender gathered three UTXO in his accounts, each with 50 BTC. One TXOut received 105 BTC, and the *change* of 44.74 BTC, after 0.26 BTC transaction fee, was sent back to the address 1NA7Mopi9b4YhuWSBrB7D4W5XsTY53N1zY, which is one of the input addresses owned by the sender. We refer to this transaction as the *3i2o-change transaction*.

2.2 Purpose of Investigation

The technical details of Bitcoins are quite complicated and tedious. Much of the complexity of Bitcoins is owed to the complex decentralized and secure ledger structure, performance requirements, and constant evolutions of the Bitcoin software and protocol to solve emerging problems. In a sense, Bitcoin is a giant software experiment. Only few people, such as cryptocurrency developers and blockchain engineers, need to know many of these low-level and tedious complexity. For IS education, most students only need to know the basic blockchain structure, which can be modeled in a high level as containing a sequence of blocks of transactions, with each transaction having possibly multiple TXIn and TXOut, in which an UTXO from a previous transaction is used as the source for a TXIn (see Figure 3). Many IS courses may only need to use this high level model.

Blockchains make very good cases for data science and analytics courses. For example, one may search using the keywords 'blockchain' or 'bitcoin' in the leading data science and analytics site Kaggle (2018), and find vibrant communities with a large collection of datasets and kernels. However, current blockchains are designed mostly as special kinds of Online Transaction Processing (OLTP) systems, but not Online Analytical Processing (OLAP) systems. Data analytics by querying the blockchain directly, such as using the reference Bitcoin's client, can be ineffective (Anh, et al., 2018). Although many Websites provide services for querying Bitcoin blockchain, they are mostly limited by their usage policies and interfaces, and can be effective only for small queries that do not process a large numbers of transactions. Furthermore, the result formats may not be suitable for analytics.

Therefore, there are much activity on extracting data from Bitcoin for storage in databases that can provide efficient accessing. For example, McGinn, McIlwraith & Guo (2018) and Spagnuolo, Maggi & Zanero (2014) both used Neo4j, an open source graphical database. In this work, we select to use SQL databases to construct examples and assignments for accessing, querying, and analyzing Bitcoin. SQL is a high level declarative language that is relatively easy to learn. Students with some database background should be familiar with it. With highly available SQL developers, it has become a de facto standard even for many non-relational databases. For example, in Big Data technologies, HiveQL is a SQL-like declarative language of Hive for MapReduce (Thusoo, et al., 2010), and Spark-SQL is a SQL dialect on top of Spark (Armbrust, et al., 2015). Similarly, cloud computing platforms also embrace SQL, such as BigQuery by Google (2018a), which supports an extension of standard SQL. Thus, our purpose is to investigate using SQL databases in IS courses for querying blockchains.

3. ACCESSING BITCOIN DATA WITH SQL

This section describes three methods we have investigated: Abe-Bitcoin, BigQuery's Bitcoin, and blockchainsql.io (bcsql). We identified a collection of query problems for Bitcoin and developed solutions on these methods as a practical way to examine them for suitability of setting assignments. The near term goal was to identify a suitable platform for assignments in an undergraduate database course.

3.1 A Local Bitcoin SQL Database

Storing the Bitcoin blockchain in a *local* SQL database allows full control and customization to satisfy diverse needs. Bitcoin blockchain is an *append-only* database in which the only change occurs about every ten minutes when a new block is created. Blocks are stored by Bitcoin Core in data files that do not change (except for the most recent evolving one) and can be parsed to populate a SQL database. There are available open source Bitcoin SQL database options, such as Abe (2018) and Bitcoin Database Generator (2018). We selected Abe because it captures more blockchain data, is more popular, and can also be used to store a number of other cryptocurrencies.

To install Abe, it is necessary to install Bitcoin Core to obtain a local copy of the blockchain first. Depending on the connection bandwidth and computer configuration, it may take a few hours to a few weeks to fully synchronize with the Bitcoin network. We selected Postgres 9.6 to install Abe because it has good performance properties and provides many modern SQL constructs. Abe is still in an Alpha version and we had to overcome a few technical issues. Eventually, the installation was complete but it took many days to do so in an old notebook.

Partial ER diagrams of the three methods are shown in Figure 4 in Appendix 1. Table 2 shows some of their basic parameters. Abe has 17 tables and 4 views. It is designed to be flexible enough to handle multiple cryptocurrencies. Many tables do not have derived columns that are computed and stored for efficiency. For example, the table txin(txin_id, tx_id, txin_pos, txout_id, txin_scriptsig, txin_sequence) stores information about transaction input. The field txin_id serves as a surrogate primary key, and tx_id and txout_id are foreign keys referencing the transaction containing the txid, and the txout used for the TXIn respectively. The other three columns are basic raw data. Users accessing a TXIn usually needs more than raw basic data and Abe uses a view txin_detail, which has 21 columns to provide contextual and summary data for the TXIn.

	Abe	BigQuery	bcsql
# tables	17	2	13
# views	4	0	0
# stored derived columns	5	0	15

Table 2 Some Parameters of the Three DB

To provide some ideas of how queries can be constructed, consider the following four problems, each related to an example transaction in Figure 1.

1. Genesis transaction: find the (Genesis) block hash from the transaction hash.
2. Pizza transaction: find the addresses and amounts of the TXIn from the pizza transaction hash.
3. Pizza-provider transaction: find the pizza-provider transaction hash, its output addresses, and amounts that used the UTXO of the pizza transaction.
4. 3i2o-change transaction: find the transaction hash of the first transaction with 3 TXIn and 2 TXOut, and also with a change going back to one of the TXIn addresses.

For reference, Appendix 2 lists the solutions to these problems using Abe. During our investigation, we found the relation schema of Abe to be relatively easy to use and we were able to construct solutions for a good collection of interesting query problems, some significantly more complicated than the four examples here. However, there was a performance issue in Abe that can be crucial when used concurrently by many students, especially since many of them are relative novices. For example, the Abe's solution for the pizza provider transaction in Appendix 2 selects from six table instances, two of which being of the table txout. It once took 143 ms to execute in an old notebook. If we replace one table instance of txout by the view txout_detail, which provides additional contextual and summary columns, the query only needs to select from three more table instances, making the query simpler. However, the execution time became 13 minutes. This is more serious in the 3i2o-change problem. The solution in Appendix 2 limits the solution space to the first 500 transactions with three TXIn and the first 500 transactions with two TXOut and hopes that the intersection of these two pools of transactions includes the result, which it does. Removing these limits make the query not able to complete in hours. Thus, students submitting non-optimized SQL queries can clog up the database. We are currently working on improving the performance of Abe. Before its performance becomes more acceptable, it is desirable to use other methods for setting the assignments.

3.2 Through Cloud Computing

Google's BigQuery is a cloud based enterprise data warehouse platform for real time data analysis using SQL that is compliant to the SQL 2011 standard and it has extensions for querying nested data (Google, 2018a). Customers are charged by the number of bytes of data processed (*scan cost*) and the first 1TB per month is free. Controlling costs by minimizing the volume of data processed of the query is a key concern in cloud computing (Google, 2018b).

BigQuery's extensions to SQL allow columns to store records and structures. Structures can be expanded into tables by using the UNNEST function, which can then be used like derived tables in the JOIN and SELECT clauses. Thus, its public Bitcoin's dataset has only two tables: blocks and transactions, with internal structures stored in columns. For examples, the many TXIn and TXOut of a transaction are stored in the columns 'inputs' and 'outputs' of the transaction respectively.

BigQuery's Bitcoin is designed mainly for fast data analytics using a columnar storage and tree architecture (Sato, 2012). Bitcoin data is filtered and selectively stored in ways to facilitate analysis for various kinds of analytic problems. It is however not designed for exploring individual transaction. The complexity for the solutions of the four problems is also higher since it does not generate a surrogate key for TXOut to easily link TXIn to TXOut. Thus, we decided not to use BigQuery as the platform for our database assignments that usually contain mostly OLTP type questions with some simple data analysis problems.

3.3 Through a Third Party Web Interface

We next investigated blockchainsql.io (2018), which contains a Web interface to submit SQL statements to query its proprietary SQL Bitcoin database. Figure 5 shows a screenshot where users can submit queries or inspect the relation schema. It uses Microsoft SQL Server and has 13 tables, with many stored derived columns to improve performance. It is sufficiently fast for the large majority of the problems we prepared for the problems, and students reported no performance issues.

Thus, with no setup and maintenance cost, reasonable performance, and ease of use, our first pilot assignment used blockchainsql.io. However, it is worthy to point out its limitations. The instructor has no control of its availability, reliability, or interface, and can only use whatever data the provider selects to provide. For example, the latest available block it provided on January 2019 was #487,853 with a timestamp of "2017-09-06 16:23:23." Thus, about 16 months of the most recent blocks were not available. Moreover, the output is in HTML and limited by the provider to 10 rows per page. It cannot be used easily as input for further data analysis. Despite these limitations, we found that blockchainsql.io is ideal for lightweight small database assignments.

4. A BITCOIN'S SQL ASSIGNMENT

We experimented with an assignment on using SQL to query Bitcoin blockchain with blockchainsql.io in an undergraduate Introduction to Database course in Spring 2018. It is homework #8 of a total of 10 assignments in the course. There was an earlier traditional SQL assignment. We gave a one hour lecture to introduce cryptocurrency, Bitcoin, and blockchain, but did not discuss blockchainsql.io as students were expected to explore it themselves. Studying and understanding existing relation schema is a course objective. Because of space, Appendix 3 shows only the core part of the assignment without the introductory narratives on Bitcoin, blockchain, and submission requirements.

The objectives of the assignments are:

1. Execute SQL statements via a third party Web interface.
2. Study the relation schema of a new application: a Bitcoin SQL database.
3. Gain insight on blockchain and Bitcoin.
4. Gain some exposure on Microsoft SQL Server. (The course mainly used MySQL.)

The assignment contains six query questions ranging from easy to beginning intermediate. Screenshots of expected output are provided with explanations. Tips and suggestions are included for the more difficult questions mainly on the difference between MySQL and MS SQL. Students need to have a good understanding of the relational schema to answer the questions, especially the more difficult ones. As a reference, the suggested solutions are shown in Appendix 4.

Before the lecture, a pre-assignment survey was conducted with 25 respondents. It shows that two students have personally invested in Bitcoin and 9 students have friends or family members invested in Bitcoin. This is a relatively high participation comparing to the general public.

In a post-assignment survey, students were asked about their perception on various aspects of the assignments in a scale of 7 (1 signifying strong disagreement, 7 strong agreement, and 4 neutral). The result is summarized in Table 3. Because of the small size of the sample, these results should only be considered to be preliminary. No quantitative analysis has been conducted.

Statement	Average
1. The assignment is useful.	5.58
2. The assignment is interesting.	6.00
3. The assignment is practical.	5.52
4. The assignment helps me gain experience on SQL execution through a Web interface.	5.65
5. The assignment exposes me to study the relational schema of a new application.	5.81
6. The assignment helps me gain insight on Bitcoin and blockchain.	5.94
7. The assignment helps me to gain experience on MS SQL Server.	5.74
8. Overall, the assignment is effective.	5.55

Table 3. Post-Assignment Survey Results

The average responses range from 5.52 to 6.00, suggesting that the assignment is relatively effective in achieving its learning objectives. The best response is on Q2 Interestingness (6.00). This suggests that a timely assignment on a confusing yet trending technology may be appealing. The response on Q3, help gaining insight in Bitcoin and blockchain, is also high at 5.94. This suggests this kind of assignments may be useful not only in a database course, but also in courses directly targeting cryptocurrency and blockchain.

We also asked two identical questions in both the pre-assignment and post-assignment surveys on the student's interest in Bitcoin and their familiarity on its technical aspects. The average responses in a scale of 5 are shown in Table 4.

	Pre	Post
Do you find Bitcoin interesting?	3.36	3.94
Are you familiar with the technical aspects of Bitcoin?	2.24	3.1

Table 4. Pre and post assignment surveys

There are marked improvements on both indicators after the assignment. However, since there were two events, the one hour lecture and the assignment, we do not know the portion of contribution from the assignment. Even with nearly no prior technical knowledge on Bitcoin, students seem to be doing fine in the assignments. The average grade for the assignment is 91.2, within the range of the average grades of 87.6 to 96.0 among the ten homework assignments. Overall, the surveys can only be considered as a pilot study but it points to the potential of using SQL to query Bitcoin blockchain as an effective learning tool.

5. DISCUSSION

As the perceived enabling technology of IoV, the next frontier in the advance of the Internet, blockchain is important in any forward looking IS curriculum. We discuss how blockchains can be incorporated into database courses as well as other IS courses in this section.

5.1 Blockchain as Database Cases

Bitcoin is not only technologically interesting, but is also a very good general case study. It is hard to find another application with such a high market capitalization and *all* transactions publicly accessible. Blockchains also make very good case studies for databases. Traditional database applications provide four basic functions of persistent data: create, read, update, and delete (CRUD). Normalization theory in relational databases aims at minimizing unnecessary data redundancy to better maintain data consistency while writing to the database (Elmasri, & Navathe, 2010; Ricardo, 2015). However, normalization may create more relations, resulting in degraded performance. Thus, when appropriate, there may be a reverse, denormalization process to improve performance. As data consistency is crucial for traditional database applications, most database courses and textbooks understandably pay much more attention to normalization than denormalization. On the other hand, in the era of Big Data, data in many newer applications is never updated or even deleted. These kinds of increasingly popular append-only databases have a strong effect of how databases should be designed and optimized but are not well treated in database courses. The Bitcoin blockchain is an excellent illustrative case study for append-only databases.

Because of space, we only discuss one other example here. Derived columns are an inadequately discussed topic in database education. As popular database textbooks, Ricardo and Urban (2015), and Elmasri and Navathe (2010) both discuss derived columns under ER-modeling in a single paragraph and both provide the classical example that 'age' is a derived column computed from the date of birth (dob). They emphasize that derived columns should not actually be stored, but computed every time when the values are needed. If, for example, age is actually stored, the functional dependency dob -> age will make the relation not in the third normal form, an indication of poor table design. This is the approach taken by Abe in which there are only five stored derived columns, all in the table Block. Instead, Abe uses views to provide derived columns (such as total inputs and total outputs in a transaction), which are computed every time. The alternate option is to actually physically store the derived columns to avoid repeated computations. The stored derived columns will need to be recomputed whenever there are changes. For write-intensive databases, that can degrade the performance significantly. Thus, most DBMS called derived columns as computed columns and they are not physically stored by default. However, an append-only databases such as Bitcoin blockchain do not have this problem as stored derived columns will never be recomputed. Blockchainsql.io have many stored derived columns and its performance is generally superior. In contrast, the views in Abe are practically too slow for many queries.

Overall, we find that Bitcoin blockchain is a very unique and good case for database education.

5.2 Uses of SQL Databases in IS Courses

The three methods have their own relative merits. Having a local database storing the blockchain is the most versatile but also requires the most work. It is also desirable to install Bitcoin Core to include the actual Bitcoin blockchain for advanced experimentations. This is especially suitable for courses on computer security, cryptocurrency, or blockchain as low level assignments can better be designed using the additional query capability beyond the blockchain. For example, we have constructed queries to identify blocks with potential security events in the blockchain (such as attempted denial of service attacks) or altcoin transactions that piggyback on the Bitcoin blockchain.

Cloud computing solutions such as BigQuery are especially suitable for courses focusing on data analytics, data science, Big Data, and cloud computing. Students will have the additional stress on constructing code that optimizes cloud computing cost, but they are standard considerations for those topics. The Kaggle (2018) website on BigQuery's Bitcoin is especially good for data analytics and data science courses as it contains an active community of rich resources and kernels.

Finally, a website such as blockchainsql.io is especially suitable as a lightweight platform for database and other introductory courses. It is just necessary for the instructor to allow plenty of time for the assignment as the site is provided by a third party in which availability is not guaranteed.

6. CONCLUSIONS

This work can be considered as a pilot study on incorporating blockchain into IS curriculum. Though limited, the initial result is encouraging. We are working on many directions to extend the project and will report the results in the future.

1. Incorporate blockchain materials and assignments into other courses, especially related to data analytics, data science, and computer security.
2. Develop assignments on using the three methods discussed.
3. Develop a more versatile and effective local platform to support blockchain and Bitcoin experiments and assignments. This includes performance refinement of Abe, extension to include other relevant datasets (such as Bitcoin's price history) and cryptocurrencies, and the uses of other database systems, especially Neo4j and MongoDB.
4. Develop tools to support our local platform.
5. Create our own blockchain applications for experimentation.
6. Conduct a quantitative study on the effectiveness of the local platform and assignments.

In summary, incorporating an important enabling technology such as blockchain into the curriculum will increase the relevance of forward looking IS programs and this paper is a contribution on this direction.

7. REFERENCES

Abe, block browser for Bitcoin and similar currencies, Retrived May 8, 2018 from https://github.com/bitcoin-abe/bitcoin-abe.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., ... & Muralidharan, S. (2018, April). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference* (p. 30). ACM.

Armbrust, M., Xin, R. S., Lian, C., Huai, Y., Liu, D., Bradley, J. K., ... & Zaharia, M. (2015, May). Spark sql: Relational data processing in spark. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (pp. 1383-1394).

Anh, D. T. T., Zhang, M., Ooi, B. C., & Chen, G. (2018). Untangling Blockchain: A Data

Processing View of Blockchain Systems. *IEEE Transactions on Knowledge and Data Engineering*.

Antonopoulos, A. M. (2017). Mastering Bitcoin: Programming the Open Blockchain. O'Reilly Media, Sebastopol, CA.

Beck, R., Avital, M., Rossi, M., & Thatcher, J. B. (2017). Blockchain Technology in Business and Information Systems Research. *Business & Information Systems Engineering*.

Bitcoin Database Generator (2018). Retrieved May 8th, 2018 from https://github.com/ladimolnar/BitcoinDatabaseGenerator.

Bitcoin.org (2018). Bitcoin Core. Retrieved May 5, 2018 from https://bitcoin.org/en/.

Blockchainsql.io (2018). Front Page, http://blockchainsql.io/. Retrieved May 3, 2018.

Delmolino, K., Arnett, M., Kosba, A., Miller, A., & Shi, E. (2016). Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In *International Conference on Financial Cryptography and Data Security* (pp. 79-94). Springer, Berlin, Heidelberg.

Detrixhe, J. (2018). Robert Shiller wrote the book on bubbles. He says "the best example right now is bitcoin." Retrieved May 5, 2018 from https://qz.com/1067557/robert-shiller-wrote-the-book-on-bubbles-he-says-the-best-example-right-now-is-bitcoin/.

Elmasri, R., & Navathe. S. (2010). Fundamentals of database systems. Addison-Wesley Publishing Company.

Ethereum (2018). Ethereum Homestead Documentation, Retrieved May 6, 2018 from http://www.ethdocs.org/en/latest/.

Foley, J., & Decker, M. (2017)The Easy Way to Create a Blockchain using Fabric Composer, Workshop Presentation, *Proceedings of the 2017 EDSIG conference*.

Google, BigQuery's front page (2018a), https://cloud.google.com/bigquery/, Retrieved May 4, 2018.

Google, BigQuery Best Practices: Controlling Costs (2018b), https://cloud.google.com/bigquery/docs/best-practices-costs

Kaggle, Bitcoin blockchain, Retrieved May 11, 2018 from https://www.kaggle.com/bigquery/bitcoin-blockchain.

McGinn, D., McIlwraith, D., & Guo, Y. (2018). Toward Open Data Blockchain Analytics: A Bitcoin Perspective. *arXiv preprint arXiv:1802.07523*.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Retrieved May 5, 2018 from https://bitcoin.org/bitcoin.pdf.

Ricardo, C. M. (2015). Databases illuminated. Jones & Bartlett Publishers.

Sato, K. (2012). An inside look at google bigquery. White paper, Retrieved May 19, 2018 from https://cloud. google. com/files/BigQueryTechnicalWP.pdf.

Spagnuolo, M., Maggi, F., & Zanero, S. (2014). Bitiodine: Extracting intelligence from the bitcoin network. In *International Conference on Financial Cryptography and Data Security* (pp. 457-468). Springer, Berlin, Heidelberg.

Tapscott, D., & Tapscott, A. (2017a). How blockchain will change organizations. *MIT Sloan Management Review*, 58(2), 10.

Tapscott, D., & Tapscott, A. (2017b). The BlockChain revolution and higher education. *Educause Review*, 52(2), 11-24.

Thusoo, A., Sarma, J. S., Jain, N., Shao, Z., Chakka, P., Zhang, N., ... & Murthy, R. (2010). Hive-a petabyte scale data warehouse using hadoop. In *IEEE 26th International Conference on Data Engineering (ICDE)*, 2010 (pp. 996-1005).

Tucker, I. (2018). Blockchain: so much bigger than bitcoin… The Guardian. Retrieved May 5, 2018 from https://www.theguardian.com/technology/2018/jan/28/blockchain-so-much-bigger-than-bitcoin.

Walch, A. (2015). The bitcoin blockchain as financial market infrastructure: A consideration of operational risk. *NYUJ Legis. & Pub. Pol'y*, *18*, 837.

Wikipedia (2018a). History of Bitcoin. Retrieved May 5, 2018 from https://en.wikipedia.org/wiki/History_of_bitcoin.

Wikipedia (2018b). List of cryptocurrencies. Retrieved May 6, 2018 from https://en.wikipedia.org/wiki/List_of_cryptocurrencies.

Yermack, D. (2015). Is Bitcoin a real currency? An economic appraisal. In *Handbook of digital currency* (pp. 31-43).

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. *In Big Data (BigData Congress)*, 2017 IEEE International Congress on (pp. 557-564).

Editor's Note:

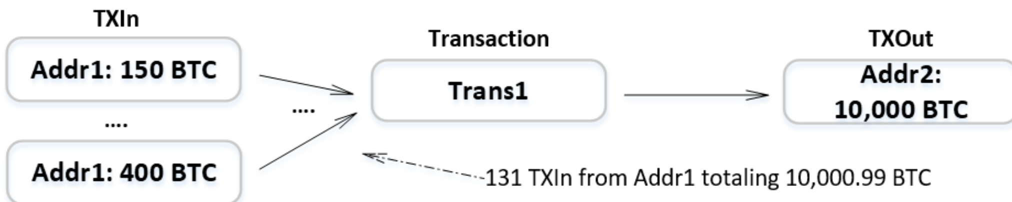
This paper was selected for inclusion in the journal as an EDSIGCON 2018 Distinguished Paper. The acceptance rate is typically 7% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2018.

Appendices

Appendix 1. Figures

Trans1: 4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b
Addr1: 1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa (TXOut)

(a) The Genesis Transaction

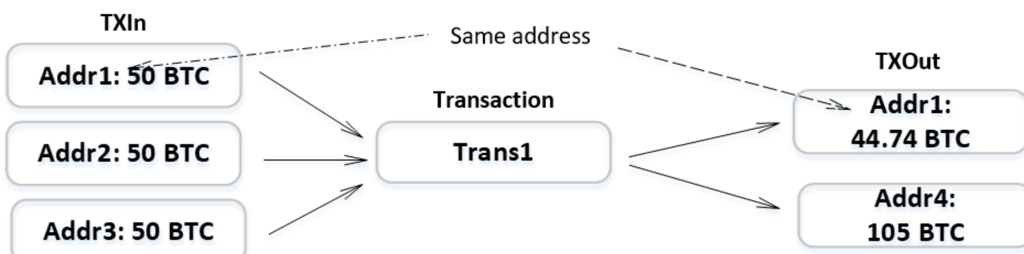


Trans1: a1075db55d416d3ca199f55b6084e2115b9345e16c5cf302fc80e9d5fbf5d48d
Addr1: 1XPTgDRhN8RFnzniWCddobD9iKZatrvH4 (TXIn)
Addr2: 17SkEw2md5avVNyYgj6RiXuQKNwkXaxFyQ (TXOut)
Transaction fee: 0.99BTC

(b) The Pizza Transaction

Trans1: cca7507897abc89628f450e8b1e0c6fca4ec3f7b34cccf55f3f531c659ff4d79
Addr1: 17SkEw2md5avVNyYgj6RiXuQKNwkXaxFyQ (TXIn)
Addr2: 1MLh2UVHgonJY4ZtsakoXtkcXDJ2EPU6RY (TXOut)
Addr3: 13TETb2WMr58mexBaNq1jmXV1J7Abk2tE2 (TXOut)

(c) The Pizza-Provider Transaction



Trans1: 418b84d7649055411d8be4e241376a93825c1d6248a304ae693060b3007a43f2
Addr1: 1NA7Mopi9b4YhuWSBrB7D4W5XsTY53N1zY (TXIn and TXOut)
Addr2: 1J27CLhDGmm3qBSiVcGxoE3evoSECUREYj (TXIn)
Addr3: 1DJ8d8gVU5VFGpSjr2AzwS9Jtg5YnyfWQD (TXIn)
Addr4: 1P3CKNyDEMRKHTDTLPqesYKSzPCo1QUCQK (TXOut)
Transaction fee: 0.26BTC

(d) 3i2o-change transaction

Figure 1. Four Interesting Bitcoin Transactions

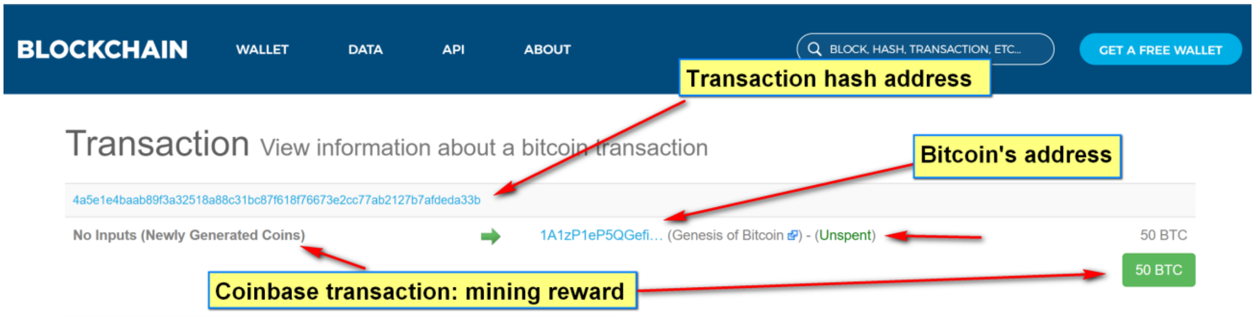


Figure 2. Information about the First Bitcoin Transaction Shown in blockchain.info

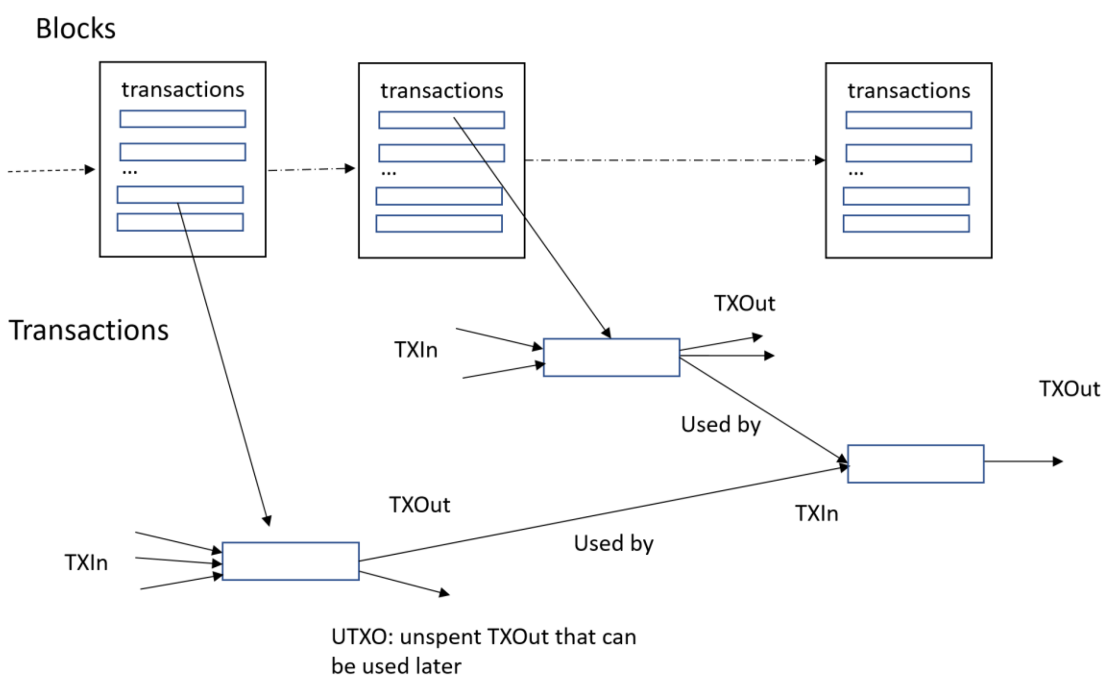


Figure 3. A High Level Model for Bitcoin Blockchain

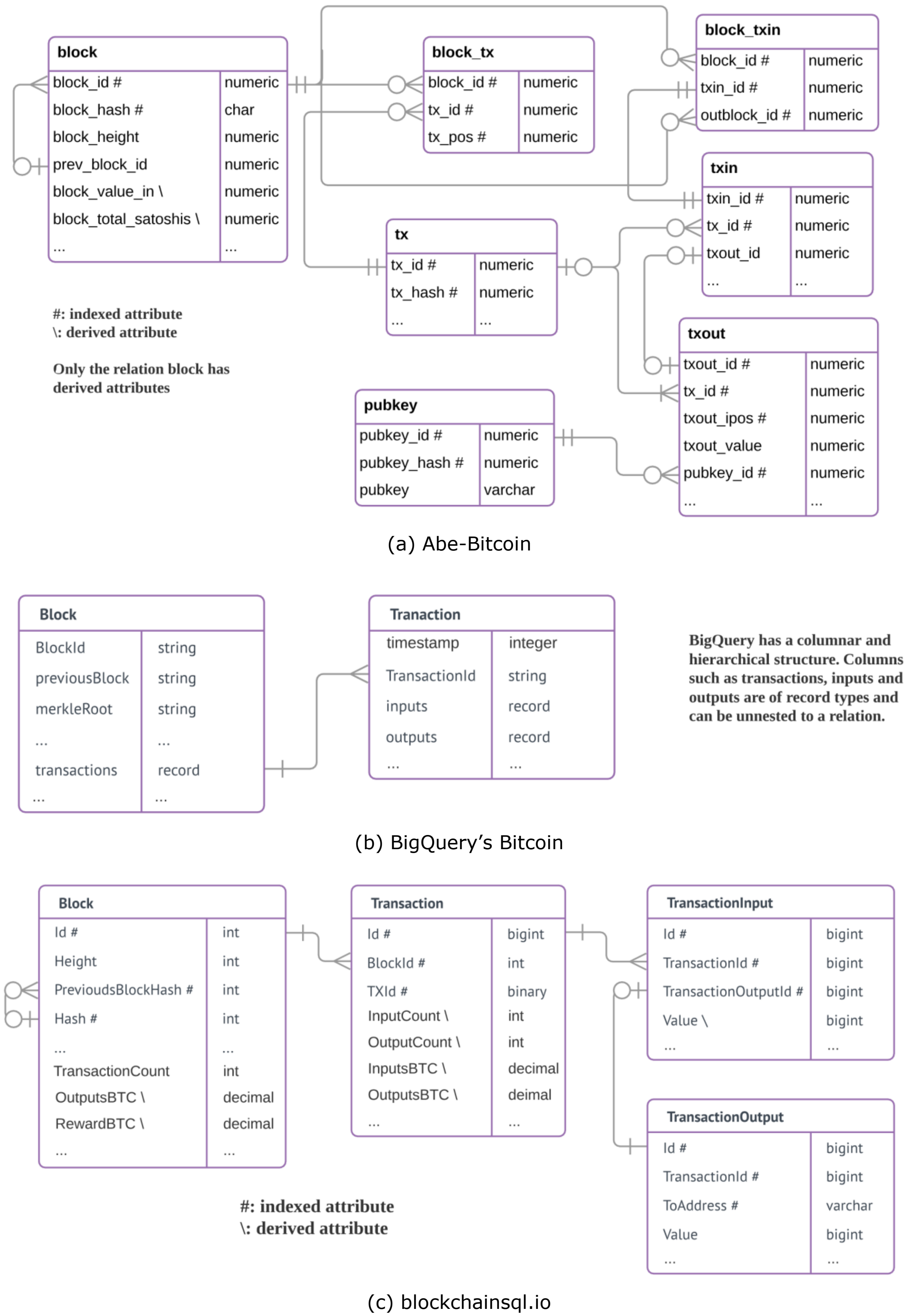


Figure 4. Partial ER Diagrams of Abe, BigQuery-Bitcoin and Blockchainsql.io

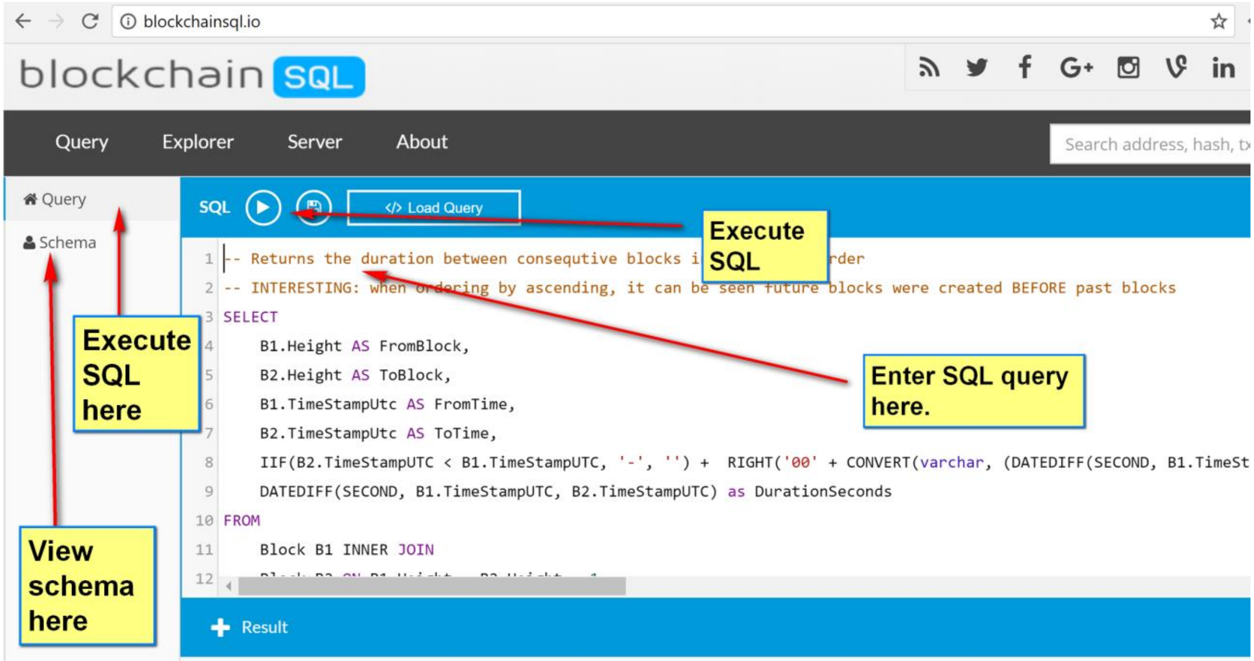


Figure 5. A Screenshot of the Front Page of Blockchainsql.io

Appendix 2 Abe's Solutions to the Four Query Problems.

1. Genesis transaction: find the (Genesis) block hash from the transaction hash.

```
select b.block_hash
from block b join block_tx bt on (b.block_id = bt.block_id)
   join tx t on (bt.tx_id = t.tx_id)
where t.tx_hash = '4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b';
```

2. Pizza transaction: find the addresses and amounts of the TXIn from the pizza transaction hash.

```
select i.txin_pos,  o.pubkey_hash, o.txout_value
from tx t join txin i on (t.tx_id = i.tx_id)
         join txout_detail o on (o.txout_id = i.txout_id)
where t.tx_hash = 'a1075db55d416d3ca199f55b6084e2115b9345e16c5cf302fc80e9d5fbf5d48d'
order by 1 asc;
```

3. Pizza-provider transaction: find the pizza-provider transaction hash, its output addresses, and amounts that used the UTXO of the pizza transaction.

```
select t2.tx_hash as tx_hash,
   p.pubkey_hash as address,
   o2.txout_pos as position,
   o2.txout_value as amount
from tx t1, txout o1, txin i, tx t2, txout o2, pubkey p
where t1.tx_id = o1.tx_id
and i.txout_id = o1.txout_id
and o2.tx_id = i.tx_id
and i.tx_id = t2.tx_id
and o2.pubkey_id = p.pubkey_id
and t1.tx_hash = 'a1075db55d416d3ca199f55b6084e2115b9345e16c5cf302fc80e9d5fbf5d48d'
```

4. 3i2o-change transaction: find the transaction hash of the first transaction with 3 TXIn and 2 TXOut, and also with a change going back to one of the TXIn addresses.

```
select distinct t.tx_id, t.tx_hash  -- depend on the fact that earlier transactions have smaller tx_id
from
   ((select tx_id from txin   -- first 500 tx with 3 inputs
     group by tx_id
     having count(txin_id) = 3
     order by tx_id
     limit 500)
     intersect
     (select tx_id from txout  -- first 500 tx with 2 outputs
     group by tx_id
     having count(txout_id) = 2
     order by tx_id
     limit 500))
     as temp  -- tx with 3 inputs and 2 outputs
join tx t on (temp.tx_id = t.tx_id)
join txin i on (temp.tx_id = i.tx_id)
join txout o1 on (i.txout_id = o1.txout_id)
join txout o2 on (temp.tx_id = o2.tx_id)
where o1.pubkey_id = o2.pubkey_id   -- output address is the change going back to an input address
order by t.tx_id
limit 1;
```

Appendix 3 A SQL Assignment on Querying the Bitcoin Blockchain

[1] Show the block information of the block with the hash address of 0x000000000000000000009769B8206EB613FBC90C607544636886E11CEB9161E33F.

Note that the hash address is the identifier of a block. Mining a Bitcoin block is more or less finding an acceptable block hash address with enough number of 0's at the beginning.

Result Executed On *2018-03-27 16:41:19 UTC* Duration *00:00:00.2867756*

ID	Height	PreviousBlockHash	Hash	E
2856743	483857	000000000000000000087C230D5CDE4C326653EC5158A1AD1F24BFDC53B7CD7F	000000000000000000009769B8206EB613FBC90C607544636886E11CEB9161E33F	1

[2] Show the height of the most recent block stored in http://blockchainsql.io/.

Result Executed On *2018-03-27 16:48:50 UTC* Duration *00:00:00.0560538*

max_height
483857

[3] Show the most recent block stored in http://blockchainsql.io/.

Result Executed On *2018-03-27 16:51:22 UTC* Duration *00:00:00.2407139*

ID	Height	PreviousBlockHash	Hash	E
2856743	483857	000000000000000000087C230D5CDE4C326653EC5158A1AD1F24BFDC53B7CD7F	000000000000000000009769B8206EB613FBC90C607544636886E11CEB9161E33F	1

[4] Average number of transactions per block in the entire Bitcoin blockchain.

avg_transaction_count
521

[5] Average number of transactions per block for every year since the blockchain was created.

year	avg_transaction_count
2009	1
2010	2
2011	31
2012	155
2013	309
2014	429
2015	840
2016	1506
2017	1814

Tips: you may find the data type of the relevant column in the relation Block by following using the 'Schema' tab in the website.

[6] Show a summary report of the transactions in the block with height 400000 with three columns:

1. "# inputs in the group": number of inputs in each group of the result. A group result is shown in one row.
2. "Number of transactions": numbers of transactions with this number of inputs.
3. "Total input Bitcoins": total inputs' BTC of transactions with this number of inputs.

# inputs in the group	Number of transactions	Total input Bitcoins
1	794	23065.99413116
2	391	900.08223074
3	185	407.30818001
4	72	120.03196718
5		
6		
10	22	55.54584783
9	14	90.2052241

In descending order of

**There are 22 transactions iwth 10 TXIn.
Their total input BTC is 55.54584783 BTC.**

Tips:

1. In SQL Server, transaction is a keyword and cannot be used directly as a table name. You will need to refer to the transaction table as [transaction].
2. MS SQL Server is stricter than MySQL in many areas. For example,

```
select s.city, s.sname, count(s.snum)
from supplier s
group by s.city
order by count(s.snum);
```

is acceptable by MySQL without any syntax error. In MS SQL Server, there are a few syntax errors:

1. s.sname is not acceptable as a select column since it is not a group by column. MySQL may just output the first s.sname in the group (which is not semantically correct as there may be many supplier names for the same supplier city). MS SQL Server produces an error.
2. In MS SQL, a column needs to have a name, and therefore you will need to use "count(s.snum) as count".
3. As a group function, count(s.snum) cannot be used in the order by clause in MS SQL.

You will need to have your SQL statement in MS SQL as:

```
select s.city, count(s.snum) as count
from supplier s
group by s.city
order by count;
```

Note that count now refers to count(s.snum) and s.sname must be removed.

Appendix 4 Suggested Solutions for the Assignment (in MS SQL of blockchainsql.io)

[1]

```
select *
from Block
where hash = 0x0000000000000000009769B8206EB613FBC90C607544636886E11CEB9161E33F
```

[2]

```
select max(height) as max_height
from Block
```

[3]

```
select *
from Block
where Height =
        (select max(height) as max_height
        from Block)
```

[4]

```
select avg(TransactionCount) as avg_transaction_count
from Block
```

[5]

```
select year(b.TimeStampUtc) as year,
        avg(b.TransactionCount) as avg_transaction_count
from Block b
group by year(b.TimeStampUtc)
```

[6]

```
select t.InputCount as "# inputs in the group",
        count(t.InputsBTC) as "Number of transactions",
        sum(t.InputsBTC) as "Total input Bitcoins"
from block b, [Transaction] t
where b.Id = t.blockId
and b.height = 400000
group by t.InputCount
order by "Number of transactions" desc
```

Integrating Big Data Analytics into an Undergraduate Information Systems Program using Hadoop

Justin Debo
jadebo@millikin.edu

RJ Podeschi
rpodeschi@millikin.edu

Tabor School of Business
Millikin University
Decatur, IL 62522

Abstract

With the emergence of big data as a strategic weapon in business, the need for hands-on activities in undergraduate courses is essential for preparing the next wave of technical talent. As the availability of programs in data analytics and data science grows based on market demands, the need for foundational technical skills is important to equip graduates for readily available entry level jobs in the field. While the available literature contains elements of application of big data into the classroom, mainstream tools like Apache Hadoop have not been readily addressed. This paper evaluates two different methods of providing students exposure to Hadoop through either an on-premise cluster or virtual machines. A curriculum is provided for students to gain hands-on experience through lab exercises, assessed through pre- and post-quizzes to test understanding. In addition, student work is assessed for application and analysis in a Business Intelligence and Big Data undergraduate course. This work contributes to the information systems (I.S.) community by providing foundational elements essential for integrating software tools such as Hadoop, Hive, and Spark into coursework.

Keywords: big data, Hadoop, business intelligence, curricula, pedagogy, data analytics

1. INTRODUCTION

With the brisk pace of technological advancement, information systems (I.S.) curricula could be rewritten annually and still not represent the most up-to-date standards of the information technology industry. The last published guidelines for model undergraduate I.S. curricula were in 2010 (Topi, et al.). It is a well-researched observation in the I.S. field that there is a gap between the skills that employers want and the skills that college students are actually learning (Wixom et al., 2014). In the data-filled world that exists today, companies seek qualified individuals who can analyze and manage large and complex data sets. According to the Bureau of Labor Statistics (2018), the job market for Computer and Information Research Scientists, on of their closest equivalents to data analytics experts, is positioned to grow by nearly 20 percent by 2026. Many schools have been slow to adapt and incorporate advanced database concepts like big data analytics and NoSQL into their curricula. Traditionally in I.S. curricula the focus remains on skills like infrastructure, programming, systems analysis, and database management.

A midwestern private liberal arts institution has positioned itself to form a collaboration between mathematics and information systems through a mathematics-focused data science major that infuses the technical data-related courses in I.S. Both data science students and I.S. students need applied skills to enter the workforce prepared to tackle the next set of technical challenges. The knowledge of theories and concepts is beneficial and necessary, but hands-on work with industry standard technology is necessary to differentiate oneself in the workplace. To gain a competitive advantage, emerging technologies should be continually introduced into I.S. curricula. This paper explores the technology behind the Hadoop open source framework for managing big data. This work also includes an evaluation of different methods of delivering Hadoop to students in a course entitled Business Intelligence and Big Data, either through a departmental server cluster or through individual virtual machines on lab computers. An outlined methodology is described to assess both learning and value from the hands-on lab exercises, and results from the first implementation of these exercises are disseminated. Ultimately, the purpose of this proposed study is to determine effective means for students to gain real experience using big data platforms and software (e.g. Hadoop, MapReduce, etc.). This research contributes to the I.S. community by aiding educators on how best to incorporate Big Data and, more specifically, Hadoop into appropriate courses.

2. REVIEW OF LITERATURE

"We are awash in a flood of data today" (Agrawal et al., 2012). Data is changing the way that entire industries function. Astronomers who used to simply take pictures of the sky are now able to analyze thousands of those pictures every day searching for new stars and galaxies. In healthcare, doctors are being aided by algorithms that sift through research data to predict illnesses and recommend treatments in time to fight previously fatal illnesses. With data from sensors, social media, and transactions, more data than ever is available to store and, more importantly, analyze. Now the challenge is how to do it. Researchers have agreed on three characteristics, often referred to as the "3 V's," that make these problems unfit for traditional relational database management systems: volume, velocity, and variety. Volume refers to the quantity of data, usually in terms of terabytes and petabytes, that companies are taking in every day. Velocity reflects the need for this data to be analyzed efficiently to provide actionable insights in real-time. The last V, variety, represents the different forms that data can take: structured, semi-structured, or unstructured. The combination of these three characteristics is what will often categorize a data set as "big data" (Coronel, 2017).

Companies all over the world are scrambling to hop on the big data train before they get left behind (Hurwitz, 2013). Previously, companies relied on large storage area networks (SAN) to store data and expensive mainframes and supercomputers for analysis. This process where all the computation is done by one machine is referred to as centralized computing. One of the biggest disadvantages of centralized computing is that it takes time to transfer data, especially when sizes reach into the petabytes. When the data are being used for real-time decisions, extra query processing time can make or break applications. Housing large quantities of data presents another challenge, and can be cost prohibitive for many companies. Increasing the storage capacity for centralized systems often means upgrading hardware, which could deplete project budgets rapidly.

Because of time and costs, businesses needed a more efficient way of storing and processing large data sets. Instead of transferring the data to centralized computers to process, in distributed processing the computing is done on the same machines where the data are stored (Hurwitz, 2013). This approach to data processing has led to the development of the Hadoop framework for big data processing. "Apache Hadoop technology is transforming the economics and dynamics of big data initiatives by supporting new processes and architectures that can help cut costs, increase revenue and create competitive advantage" (IBM, 2014).

As Prajapati (2013) states, Hadoop is made up of its storage system, Hadoop Distributed File System (HDFS), and its distributed processing framework (MapReduce). "Scalability and availability are key traits of HDFS, achieved in part due to data replication and fault tolerance" (Holmes, 2015) as seen in Figure 1 below. HDFS stores data sets by breaking them up into "blocks" and replicating these blocks to be stored on a cluster of servers. HDFS functions on the principle of "Horizontal Scaling" (Warden, 2011). As opposed to traditional vertical scaling, where a company would simply buy bigger servers to store more data, in horizontal scaling when storage needs increase, companies instead buy more commodity computers and add on to the Hadoop clusters. This process, combined with the data replication, makes HDFS incredibly fault tolerant, since anytime one of the computers in the cluster (a "node"), fails, it can be easily replaced and the data that are stored on it can be copied from the replicated blocks back onto the new node.

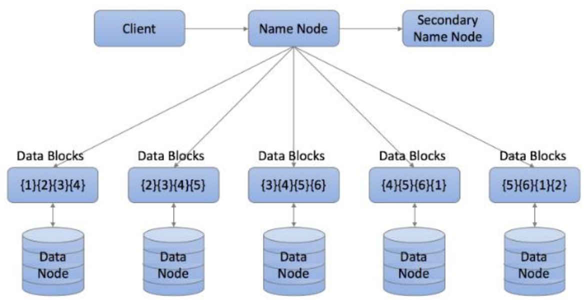


Figure 1. HDFS model

The MapReduce framework partners with HDFS to allow companies to analyze data from different sources that "could take days or longer using conventional serial programming techniques…" (Holmes, 2015). MapReduce is made up of two operations, a map function and a reduce function. The mapping determines how the cluster is going to divide the work and delegate data processing, per block, to computers with available capacity. The reduce function reassembles the data after processing is complete as seen in Figure 2 below. With its ability to store and analyze large data sets on commodity hardware, Hadoop has become an integral part of data analytics at many industry leading companies.

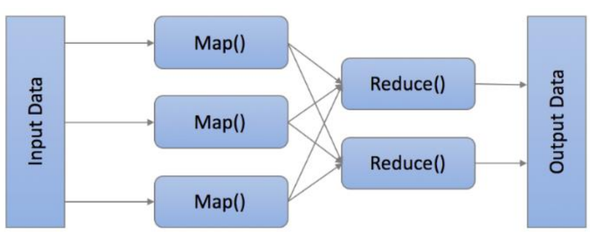


Figure 2. MapReduce model

The current research question is, how can this technology be applied in a classroom setting to prepare students for work in the real world? According to the findings of the Business Intelligence Congress 3, as described in Wixom et al. (2014), "Employers are not satisfied with the practical experience of university graduates" and "Demand for BI/BA students continues to outpace supply." As early as 2003 (Lu, Bettine), the need for I.S curriculum modernization in the areas of business intelligence and business analytics has been recognized. I.S departments have been working hard and fast to incorporate business intelligence into their curricula, to try and satisfy the industry demand for experienced graduates. Mrdalj (2007) writes about the creation of an applied, graduate level, business intelligence course at Eastern Michigan University. More recently, Podeschi (2015) describes the use of an industry software, QlikView, to construct an experiential learning environment in an undergraduate level business intelligence class. The use of industry standard technology is vital to successful hands on learning. Molluzzo and Lawler (2015) proposed a "Concentration Curriculum Design for Big Data Analytics for Information Systems Students". In this paper, the authors discuss the different technologies they plan to incorporate into their data science concentration, including Hadoop and MapReduce. They again discuss the industry need for experts who have the knowledge to work with big data projects, and the lack of supply of such graduates. While previous research includes the discussion of these big data topics in the classroom, there is still a lack of evidence that demonstrates the incorporation of hands-on learning related to tools such as Hadoop, MapReduce, and so on. The research does agree, however, on the need to incorporate industry standard technology to prepare students for the real world.

3. COURSE OBJECTIVES

Students taking the Business Intelligence and Big Data course are part of the I.S. major, data management certificate, or data science concentration from mathematics. Students in the I.S. program are grounded in the theoretical areas of programming, system analysis and design, relational databases, and I.T. infrastructure. Through the major, students learn hands-on skills in both business and technology that prepare them for jobs managing data, developing web applications, securing systems, and analyzing systems. Prior to this course, it is expected that students have an intermediate knowledge of spreadsheets, an introductory knowledge of SQL, a broad understanding of the fundamentals of information systems, and have taken at least one statistics course. This Business Intelligence (BI) course introduces and builds on the concept of data warehousing and modeling data for analysis rather than for transactional and operational processes. This course provides students with hands-on experience in data warehousing, data analytics, and executive dashboards through real-world data sets and applications. Students are exposed to a variety of software tools throughout the course including: Oracle Database, QlikView, R, and as introduced in this paper, Hadoop and MapReduce. The primary learning objectives of the course are:

- Understand the strategic importance of business intelligence and data analytics.
- Understand the difference between descriptive, prescriptive, and predictive analytics.
- Design and develop a data warehouse based on data needs and user requirements.
- Extract, transform, and load operational data into a data warehouse.
- Build a business intelligence application for dashboarding, analysis, and reporting.
- Interpret data into informed decisions for recommendation.

4. LAB ENVIRONMENT

Prior to developing lab exercises for students, two different platforms were evaluated: departmental servers using manual installation and configuration (on-premises) or virtual machine appliances provided by Cloudera for students to gain hands-on experience. The first approach involved installation and configuration of Hadoop and its associated tools on two used HP servers. Cloudera QuickStart virtual machine appliances, with Hadoop and tools pre-installed, were also evaluated. These two approaches provide different benefits and costs to the institution, the faculty, and student learning.

On-premises Hadoop Cluster

To build the on-premises option, the lack of computing resources needed to be addressed by acquiring sufficient hardware to support a cluster-computing environment. Through corporate donations and departmental funds, the program acquired two used HP servers and sufficient hard disk storage and memory. Using multiple servers on-premises allows students to observe how distributed computing works in a small environment with one server as the name node and the other as a data node.

Hadoop is built to run on a Linux operating system, so the decision on which operating system to use was first. Red Hat Enterprise Linux (RHEL) is the industry standard for large scale server infrastructures (Gillen, 2017). A significant number of industry companies use RHEL for their server architecture. Because funding was neither available nor warranted for RHEL licenses, CentOS 4.8.5 was selected for its price (free), stability, active support community, and close code base to RHEL. Hadoop 2.7.2 was the latest stable version at the beginning of the project.

Cloudera QuickStart Virtual Machine

The other architecture evaluated was a virtual machine appliance provided by Cloudera, a value-added Hadoop vendor. In this scenario, students have their own Hadoop single-node cluster in order to follow prescriptive labs, allow for iteration, and even experience failures. While Cloudera does not provide free licenses for their full enterprise distribution of Hadoop for academic use, they do offer a free sandbox virtual machine called Cloudera QuickStart. Cloudera QuickStart allows users to download virtual machines (VMs) or Docker images that come pre-installed with Hadoop and a set of applications that can be built into the Cloudera Distributed Hadoop (CDH) platform as seen in Image 1 in Appendix A. This platform gives students the ability to build their own isolated Hadoop environment to use in class, as well as develop and test applications they wish to eventually run on the full cluster environment.

Both environments allow students to see how the Hadoop ecosystem functions. Building an on-premises cluster for student-use takes a significant amount of effort for system and network configuration along with creating a cluster. This environment would be beneficial for students who are interested in pursuing careers as server or system administrators as more of the individual components are exposed. However, a department should go through a proper analysis of available resources for not only installation and configuration, but also maintenance and support. While beneficial to understand the architecture, the cost of time may be too burdensome for departments with fewer resources. Cloudera QuickStart, however, provides a quick and easy way for students to deploy an environment with the appropriate tools for understanding the fundamentals. Exercises and labs, in this instance, are likely to be smaller in scale, as hard disk space and memory capacity become concerns. For purposes of getting students early exposure to Hadoop and its tools, Cloudera is a better suited platform for designing introductory labs. While more resource intensive, an on-premises server cluster is likely more appropriate for independent research or client projects where additional computing resources like CPU, memory, and disk space are necessary.

6. METHODOLOGY

After determining that a department Hadoop cluster was too administratively burdensome to manage, Cloudera QuickStart was used for introductory labs for students to gain initial exposure to Hadoop and its associated tools such as Hive and Apache Spark. Learning the Hadoop ecosystem, the available tools, and its terminology (as depicted in Figure 3 below). These lab exercises took take place during the last third of the semester. By this point in the course, students will have been exposed to traditional forms of business intelligence through data warehousing concepts along with descriptive, prescriptive, and predictive analytics.

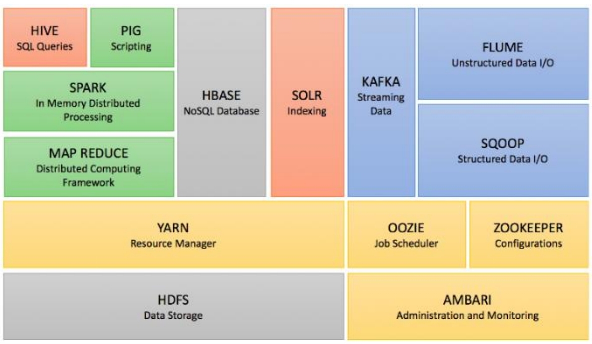


Figure 3. Hadoop ecosystem

This module of the course begins with an overview of big data, Hadoop and its associated tools, along with appropriate terminology and definitions. Students can gain further understanding through cases, and in-class demonstrations. More detailed Hadoop, HDFS, Hive, and Spark material can be supplemented through additional videos and content produced by Cloudera's OnDemand training, such as the Cloudera Essentials for Apache Hadoop course. Once a solid foundation is achieved, students will then go through lab exercises, as described, to gain an exposure to Hadoop and its associated tools.

The primary learning objectives for this module that support the course objectives are to: 1) recognize and recall definitions and terminology related to big data and Hadoop; 2) understand the key difference between traditional and NoSQL databases and the use cases for each; 3) understand the key advantages of distributed data storage and processing; and 4) gain an introductory exposure to Hadoop, HDFS, Hive, and Spark. These learning objectives intentionally map to Bloom's Revised Taxonomy as seen in Table 1 (Krathwohl, 2002). As a result, students move from lower to higher order thinking skills throughout the course of the module. From the knowledge dimension, students should be able to gain both factual and conceptual knowledge about these elements, and through the lab exercises, gain procedural knowledge of what steps to take and when, specifically, to take them based on a given problem.

Learning Goal	Bloom's Revised Taxonomy
1. Recognize and recall definitions and terminology related to big data and Hadoop	Remember, Understand
2. Understand the key differences between traditional and NoSQL databases, and the use cases for each.	Understand, Analyze
3. Understand the key advantages of distributed data processing through HDFS	Remember, Understand
4. Gain an introductory exposure to Hadoop, HDFS, Hive, and Spark	Apply, Create

Table 1. Module learning goals mapped to Bloom's Revised Taxonomy

Lab Exercises

With every student having their own full environment through Cloudera QuickStart, either on their personal machine or in the computer lab, it became possible to spend more time on labs and less time getting students an environment ready. With Cloudera QuickStart, instead of narrowing the curriculum to look at just Spark and HDFS, it could incorporate a holistic education on the Hadoop ecosystem.

As part of the lab exercise, students, individually, paced through the Cloudera QuickStart "Getting Started with Hadoop" tutorial (https://www.cloudera.com/developers/get-started-with-hadoop-tutorial.html). The tutorial familiarized students with common Hadoop tools such as HDFS, Hue, Spark, Hive, and Impala. The tutorial is positioned as a case for a fictional company called DataCo where students are asked to answer key questions about data for DataCo management. Students are required to use the virtual machine to ingest data using Sqoop and Hive, query relational data using SQL in Hue, correlate structured data with unstructured data using log files using Hive and Impala, and analyze real-time data using Flume and Morphline. While the tutorial is prescriptive, it does provide a ready-to-use class exercise for students to gain exposure to the variety of tools used in a Hadoop environment. Students were required to complete the tutorial and document their journey through screen captures and write a narrative describing what they were doing while connecting it back to the course material. At the end of the tutorial, students were asked what they liked the most and least about the lab, and specifically, how the tutorial assisted them in better understanding Hadoop and its associated tools.

Assessment

To evaluate the effectiveness of the first implementation of the curriculum, students will be assessed on the first two domains of Bloom's Revised Taxonomy, remembering and understanding, through administering pre- and post-quizzes. The quiz is designed to measure the comprehension and retention of the material at a conceptual level. Quiz questions were adopted from the instructor materials that accompanied the textbook by Sharda, Delen, and Turban, used in the course (2018). The questions were selected for their relevant content from the Big Data module and having been peer reviewed from academia and industry. The quiz was administered at the beginning of the big data unit to establish a baseline of the students' preexisting subject knowledge. The same quiz was administered after the unit to provide evidence of the impact of the coursework.

Application and analysis of key tools (HDFS, Spark, Hive and HBase) will be assessed through the hands-on labs as described in the methodology. During key points of the labs, students were asked short essay-style and reflective questions requiring them to connect the class content to the lab exercises, and ideally, application. The rubric for the labs will focus on the successful completion of the prescribed exercises, as well as the students' thorough responses to the prompts.

7. RESULTS

The goal of these hands-on lab exercises is for students to walk away with more than just exposure to current tools in the big data space. Overall, as gathered from their reflective responses, that they most liked having their hands on the technology as opposed to watching a video or watching a demonstration in class. Students learned the connection between the various Cloudera services as well. Furthermore, students felt at ease knowing that many of the tutorial exercises relied upon the SQL skills they acquired from previous coursework. One student reflected: "I enjoyed the entire process of moving the mass of unstructured and semi-structed into fields of a table and then further distilled so that it was SQL-usable—it felt like I learned something extremely useful and versatile." From the understanding perspective, students experienced a concrete example of how to use Hadoop to analyze large data sets, and connected it to the concept of extract, transform, load (ETL). One student commented that "the tutorial helped me learn the differences between what each application was able to do. For example, I learned that Sqoop is used to pull data into Hadoop, Hive is used for creating tables, and Impala is used to query data. Each application has an important part to play, and when putting them all together, we were able to accomplish our goals."

The majority of the negative reactions to the lab were technical in nature related to limitations of personal or lab computers. Students didn't like the amount of time it took for Cloudera services to start up or data to load. In addition, some students stated frustrations with memory allocation and configuration.

Ideally, after comparing results from the pre-assessment to the post-assessment, students should perform better on the post-assessment. Unfortunately, due to the small class size (n=9), analysis of the pre- and post-assessment did not provide a reliable data set to assess the lower-order thinking skills according to Bloom's Revised Taxonomy. It is anticipated that the collection of these assessments over time will lead to more insightful results.

In general, reflective responses demonstrated a connection between the knowledge and the application through the hands-on labs. In connection with the previously stated learning goals and Bloom's Revised Taxonomy, student reflections demonstrated higher-order thinking skills of understanding and analysis relative to learning goal 2) understand the key differences between traditional and NoSQL databases, and the use cases for each. Furthermore students applied and created through learning goal 4) gain an introductory exposure to Hadoop, HDFS, Hive, and Spark (Table 1).

The quantitative analysis of the pre- and post-assessment instrument, given a larger sample size or over time, would have provided an additional lens to evaluate the effectiveness related to Bloom's Revised Taxonomy on the lower-order thinking skills, and provided a way to measure learning goals 1) Recognize and recall definitions and terminology related to big data and Hadoop; and 3) Understand the key advantages of distributed data processing through HDFS (Table 1). The experiences gathered from students through this study, however, have provided valuable information on how to structure hands-on labs related to big data in the future, with implications for contributing to best practice pedagogy in big data.

8. DISCUSSION AND CONCLUSIONS

As big data technology continues to mature, the tools emphasized in the curriculum will need to be reviewed on an ongoing basis to assess industry relevance and currency. Furthermore, the engagement of industry experts, some of them alumni, and their input through such mechanisms as advisory boards will be beneficial to keeping the content relevant.

This paper provides a good point of reference for information systems programs looking to incorporate big data skills into their curricula. The details of this learning module should serve as a template for developing new labs based around Hadoop. While the content is subject to individual preference, the keys to making labs like these effective are to build them upon preexisting skill sets and make them easily accessible to students with limited domain knowledge. Utilizing the Cloudera QuickStart VMs removes a number of technical knowledge barriers, has lower administrative overhead, and provides a no-cost solution to giving students hands-on Hadoop experience. Students had little difficulty importing the QuickStart VM into the preferred hypervisor or configuring the settings. However, some students ran into memory limitations. While the documentation states that the VM will perform with 8gb of RAM allocated, providing for more memory is recommended. Students who allocated up to 12gb of RAM to the VM had fewer issues starting services and shorter load times for large data sets. For these reasons, Cloudera QuickStart provides the most cost-effective and efficient solution for labs in a classroom setting.

For institutions interested in the server cluster environment, it is important to note the issues involved with sustainability. For the cluster to be sustainable, there needs to be a way in which it can be maintained, updated, reconfigured and, if necessary, rebuilt. It requires dedicated personnel who have some knowledge of the Hadoop ecosystem and thorough documentation for reference to manage the ongoing maintenance, updating, and troubleshooting that would be necessary for an environment like this. A proposed solution is to create a sustainable fund, through a combination of donor funding and third-party business investment, to employ a student to oversee the environment and perform the necessary maintenance. It is, however, unrealistic to expect that every student system administrator will have the time or knowledge to rebuild the environment in the event of a system failure, even with the best documentation. A necessary extension to this solution is to create an automation script, using Python, that will allow student administrators to easily deploy new servers or to rebuild the system.

Information systems educators have wrestled with similar issues in the past such as how to provide students with hands-on experience in areas of programming, database, and so on. It is anticipated that the results of this research study, and its continuation, will help educators better align applied big data and related courses with other hallmark information systems courses such as system analysis and design, database, and enterprise architecture.

8. NOTES

1. Cloudera's QuickStart VM is available at https://www.cloudera.com/downloads/quickstart_vms/5-13.html
2. Getting Started with Hadoop Tutorial is available at: https://www.cloudera.com/developers/get-started-with-hadoop-tutorial.html
3. Hadoop cluster specifications: Two HP DL380 G7 servers, 1.4 terabytes of hard disk storage, and 128 gigabytes of RAM.
4. CentOS is available for free at https://www.centos.org/

9. REFERENCES

Agrawal, D., Bernstein, P., Bertino, E., Davidson, S., Dayal, U., Franklin, M., … Widom, J. (2012). Challenges and opportunities with big data [White Paper]. Retrieved April 17, 2017 from https://www.purdue.edu/discoverypark/cyber/assets/pdfs/BigDataWhitePaper.pdf

Doherty, C., Camina, S., White, K., & Orenstein G. (2016). The path to predictive analytics and machine learning. Sebastopol, California: O'Reilly Media, Inc.

Gillen, A., Marden, M., & Perry, R. (2017). The business value of Red Hat Enterprise Linux [White paper]. Retrieved May 9, 2018, from https://www.redhat.com/en/resources/idc-whitepaper-business-value-red-hat-enterprise-linux

Guttag, J. V. (2016). Introduction to Computation and Programming Using Python. Cambridge, MA: The MIT Press.

Holmes, A. (2015). Hadoop in practice. Shelter Island, NY: Manning Publications Co.

Hurwitz, J., Nugent, A., Halper, Dr. A., & Kaufman, M. (2013). Big data for dummies. Hoboken, NJ; John Wiley & Sons, Inc. IBM. (2014). Big data integration and Hadoop [White paper]. Retrieved March 18, 2017, from https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=IMW14791USEN

Krathwohl, D. R. (2002). A Revision of Bloom's Taxonomy: An Overview. Theory Into Practice, 41(4), 212.

Molluzzo, J.C., & Lawler, J.P. (2015). A Proposed Concentration Curriculum Design for Big Data Analytics for Information Systems Students. Information Systems Education Journal, Volume 13 No. 1, 45-52.

Lu, Y., Bettine, J. (2003). Data Mining: An Experimental Undergraduate Course. Journal of Computing Sciences in Colleges. 18(3), 81-86.

Mrdalj, S. (2007). "Teaching an Applied Business Intelligence Course." Issues in Information Systems, 8(1), 134-138.

Podeschi, R. (2015). Experiential Learning using QlikView Business Intelligence Software. Information Systems Education Journal, 13(4) pp 71-80.

Prajapati, V. (2013). Big Data Analytics with R and Hadoop. Birmingham, UK: Packt Publishing Ltd.

Sharda, R., Delen, D., Turban, E. (2018). *Business Intelligence, Analytics, and Data Science*. New York, NY. Pearson.

Topi, H., Valacich, J., Wright, R., et al. (2010). IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. Association for Computing Machinery and Association for Information Systems.

United States Bureau of Labor Statistics. (2018). *Computer and Information Technology Occupations*. Available from https://www.bls.gov/ooh/computer-and-information-technology/home.htm

Warden, P. (2011). Big data glossary. Sebastopol, CA: O'Reily Media Inc.

Watson, H. (2008). "Business Schools Need to Change What They Teach." Business Intelligence Journal, 13(4), 4-7.

Wixom, B., Ariyachandra T., David, D., Goul, M., Gupta, B., Iyer, L,. … Turetken, O., (2014). "The Current State of Business Intelligence in Academia: The Arrival of Big Data." Communications of the Association for Information Systems. 34(1), 1-13.

Appendix A

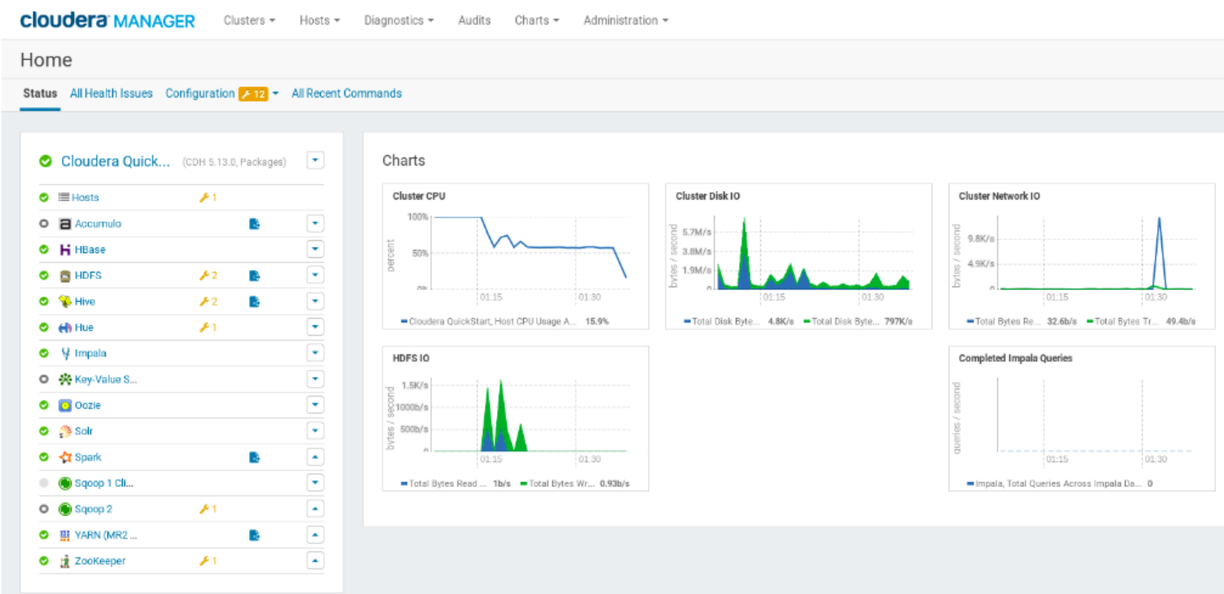


Image 1. Cloudera QuickStart VM

Toward Visualizing Computing Curricula: The Challenge of Competency

Leslie J. Waguespack
lwaguespack@bentley.edu
Computer Information Systems
Bentley University
Waitham, MA 02452

Jeffry S. Babb
jbabb@wtamu.edu
Computer Information and Decision Management
West Texas A&M University
Canyon, TX 79016

Abstract

Amidst academic societies and agencies that accredit computing education there is a growing enthusiasm to reexamine the efficacy of the traditional model of curricular description that focused on areas of knowledge. The knowledge model informed the architecture and design of programs of teaching and learning in post-secondary, degree-granting institutions. The incipient enthusiasm for change draws on a vocational heritage focusing on job performance together with outcomes-based and task-centered learning and assessment, competency. Competency's emergence is fueled by a waning confidence in the cost-effectiveness of college-based education, an industry perception of a persistent short-fall of technology-savvy hiring prospects, and the efforts of governments worldwide encouraging the alignment of public education with economic and workforce policy. The competency model represents the consequence of learning as a blend of knowledge, skills, and disposition – "*knowing* what," "*knowing* how," and "*knowing* why." The *knowing* is task-focused both as the learning *in doing* and the assessment as demonstration *in doing*. Coincidentally, ACM and IEEE have undertaken to reprise CC2005 with the goal of an online interactive curriculum modeling tool for comparing and exploring curricular guidelines and academic programs. We explore the competency-based curricular approach situated in a) the history of computing curricula standardization, b) its heritage in education originating with clinical and professional disciplines, and c) its implications on the CC2020 project's aspirations of designing a "tool" to facilitate current and future competency-based computing curricula development.

Keywords: Computing Competency, Curricular Guidelines, Knowledge (Areas, Units, Learning Outcomes), Computing workforce preparation

1. INTRODUCTION

This paper takes a brief look back at computing curricula and their constitution and a look forward in light of the growing and earnest interest in competency as a design medium of education. ACM and IEEE partnering with sibling computing associations are spearheading an ongoing inventory and forecast of computing curricula development, *Computing Curricula 2020*. (See www.cc2020.net.) The CC2020 effort is a response to an accelerating advance of technology and evolving perspectives on the valuation of educational outcomes of computing programs. Particular to that effort, this paper sketches a framework for curriculum description that incorporates and normalizes the structure and intra-connectivity of computing theory and practice. There has never been a rubric for a top-down description of curriculum spanning computing education. The framework is a necessary foundation for a key CC2020 project goal, the design of an online visualization tool capable of both representing and comparing computing guidelines and programs to inform and advance computing education in the 2020's and beyond. As the authors are presently engaged in this ongoing design, our goal here is to report and explore the challenges intrinsic to this undertaking.

2. BRIEF COMPUTING CURRICULA RETROSPECTIVE

Since 1968, professional computing communities have invested in developing guidelines that chart a path for computing education in degree-granting institutions, and to some extent, the entire community of practicing professionals (Longenecker, Feinstein, Babb, 2013). By and large, the various curricular guidelines have focused on the delivery of subdiscipline-specific, fact-based information aligned with a scientific and technically-rational model of instruction. Sub-disciplines of computing have evolved generally independent of one another creating de facto silos of perspective on computing albeit sharing significant overlaps of theory, technology, methodology, and professional practice.

The traditional sub-disciplines of computing are codified in the collection of baccalaureate level, curriculum guidelines published under the sponsorship of ACM and IEEE with various partners over the past couple decades: Computer Engineering (2004, 2016), Computer Science (2001, 2008, 2013), Information Systems (1997, 2002, 2006, 2010), Information Technology (2008, 2017), Software Engineering (2004, 2014), and Cybersecurity (2017). (All guidelines are available at www.acm.org/education/curricula-recommendations.). As of this writing projects are underway for new and/or updated sub-discipline guidelines for data analytics, artificial intelligence, and information systems.

A baccalaureate degree in computing has been the traditional flagship of credentialed entry to the computing profession. In the last decade or so, instructional approaches focusing on a much narrower conception of professional preparation, "computing" bootcamps, have arisen as an alternative and to some extent, have become challengers to the traditional baccalaureate programs (Waguespack, Babb, Yates, 2018). Computing education is also widely accessible over the internet in both tuition and tuition-free models without the traditional "brick and mortar" institution (e.g. Coursera, 2018). The pace of computing technology's expanding influence on society and the growth of related scientific and technological information continues to accelerate; as does the growing need for workers in the computing domain. The time is nigh to take stock of the breadth and width of computing education both for orientation and navigation with a focus beyond 2020, hence the impetus for the CC2020 project.

3. CC2005 CURRICULUM VISUALIZATION

The Association for Computing Machinery, ACM, and the Institute for Electrical and Electronics Engineering, IEEE, have worked consistently to normalize the structure and evolution of computing education. The series of published curricular guidelines for particular computing disciplines, as well the mapping of the overall landscape as in Computing Curriculum 2005 (CC2005) are the current, standard references for computing education (Shackelford, McGettrick, Sloan, Topi, Davies, Kamali, Cross, Impagliazzo, LeBlanc, & Lunt, 2005).

In that CC2005 report (the most recent and comprehensive cross-discipline analysis), the task force created graphic characterizations of "what students in each of the disciplines typically do after graduation." Each discipline was portrayed on a field of credible, professional capability as a "footprint" of proficiency gained by completing the respective academic program. (See Figure 1.)

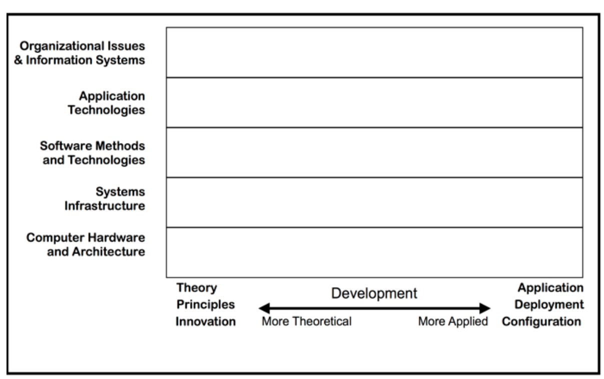


Figure 1 - CC2005 Field of Computing Competency

In that 2005 overview report, the five computing disciplines (CS, CE, IT, IS, and SE) were represented in individual "footprint" diagrams. Each diagram was a plausible depiction negotiated by a committee of curricular "experts." Figure 2 depicts IS education in 2005 as an example (Shackelford et al., 2005).

The "footprint" of baccalaureate proficiency lies on a field delineating computing activity ranging on the Y-axis from hardware issues on the bottom to organizational policy and information management at the top. The X-axis depicts the far right as purely *applied* involvement in computing activities while to the left is purely *theoretical* engagement with computing topics. In addition to the "footprint" representations, the CC2005 report also provided a tabular representation of the plausible relative emphasis of knowledge areas as documented among the respective curricular guidelines. (See Appendix A.)

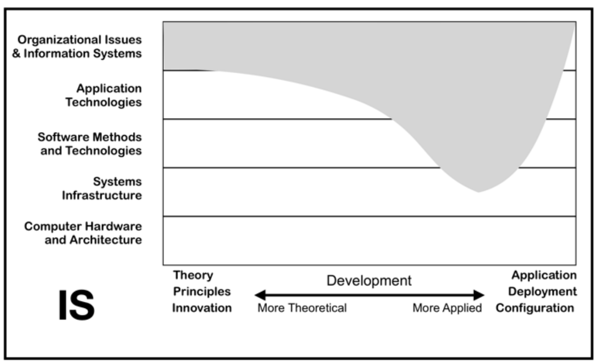


Figure 2 - Competency Target of IS (2005)

When CC2005 was published, the intention was to revisit the computing landscape every five or so years to both chart the "status-quo" and to offer a reflection on the emerging demands on computing education owing to advancing technology and society's expanding dependence on computing. However, that intention has waited nearly fifteen years for the mantel to be taken up again in CC2020.

Now underway, the CC2020 project aspires to reprise the effort to visualize the computing landscape. But in this attempt, rather than depictions of "expert" opinion as offered in CC2005, the goal is to derive the visualizations of the computing disciplines based upon actual descriptions in the respective published curriculum guidelines. Those CC2005 depictions, although speculative, have proven invaluable as approximate contrasts of scope and focus among the various curricula. But, the continuing addition of new disciplinary variations fuels the need to do more than speculate. "More" entails a theory-grounded articulation of the spectrum of computing education, an instrument to compare and contrast degree programs, guidelines, and disciplines; and an empirical, "algorithmically" rational visual representation of curricular specifications. While automating a visualization of the computing curricula as conceived in 2005 presents taxonomical and epistemological challenges in itself, the most recent curricular guidelines (MSIS2016 and IT2017) have undertaken a new descriptive strategy aligned more closely to training, certification, and job description that is more familiar to the domain of human resources. This strategy, *competency-based*, represents an educational goal of instilling students with a combination of knowledge, skills, and disposition amenable to validation using task-specific proficiency assessment.

In the spirit of the "footprint" diagrams of baccalaureate capabilities that grace the CC2005 report, CC2020 envisions an instrument able to:

a) visually represent curricular guidelines or programs for exhibition and analysis,
b) permit navigation and inspection of such guidelines or programs at varying levels of scale / detail,
c) detect commonalities and differences between or among curriculum descriptions and display same visually,
d) accommodate the drafting of curriculum guidelines (i.e. any proposed or published subdiscipline of computing) or any existing or proposed program of study in computing as an aid to prototyping and development,
e) support the manipulation and versioning of curriculum descriptions,
f) (implicit in a thru e) facilitate a repository for the cumulation and evolution of a taxonomy of computing to support cross comparison and future curricular development.

The CC2020 project team refers to this visualization capability as "the tool." The goal of this paper is to explore the challenge that CC2020 has accepted and conceptualize a framework for incorporating *competency* in the visualization effort.

4. KNOWLEDGE VS COMPETENCY FOCUS

All the published baccalaureate computing curricula in 2005 were conceptually grounded in a listing of knowledge areas, knowledge units, and learning outcomes (KA-KU-LO). (See

Appendix A, KA examples found in the CC2005 report.) Thus, the visual representations plausibly exhibit a degree of comparability across subdisciplines. The arrival of IT2017 (IT2017, 2017), heralds a shift in specification strategy, less conformant to the basis of KA-KU-LO – specifically due to its emphasis on *competency*. (See Appendix B comparing the published computing curricula that are the current focus of CC2020.)

The IT2017 project is the first of the ACM, IEEE baccalaureate curriculum projects to embrace competency as the primary characteristic of curriculum definition. MSIS2016 introduced competencies earlier but, also included prototypical course descriptions reminiscent of KA-KU-LO. In some ways the competency approach is more a facet of labeling than a dramatic shift in curricular description. Learning outcomes have been a prominent feature for some time (USDoE, 2018).

The learning outcome concept is key to the shift in education from a paradigm concerned with providing instruction to a paradigm of producing learning (Barr, 1995).

Table 1 – Six facets of learning transfer

Explain	Learners make connections, draw inferences, express them in their own words with support or justification, use apt analogies;
Interpret	Learners make sense of, provide a revealing historical or personal dimension to ideas, data, and events; interpretation is personal and accessible through images, anecdotes, analogies, and stories; turn data into information; provide a compelling and coherent
Apply	Learners use what they have learned in varied and unique situations; go beyond the context in which they learned to new units, courses, and situations, beyond the classroom.
Demonstrate Perspective	Learners see the big picture, are aware of, and consider various points of view; take a critical and disinterested stance; recognize and avoid bias in how positions are stated.
Show Empathy	Learners perceive sensitively; can "walk in another's shoes;" find potential value in what others might find odd, alien, or implausible.
Have Self-Knowledge	Learners show metacognitive awareness on motivation, confidence, responsibility, and integrity; reflect on the meaning of new learning and experiences; recognize the prejudices, projections, and habits of mind that both shape and impede their own understanding; are aware of what they do not understand in a specific context.

But, the prominence of competency further emphasizes that *fact knowledge* does not sum all the *knowing* sufficient to equip a practicing professional. Competency is a familiar term in the domains of education usually classified as *training* and *job* performance assessment. Competency is identified with job recruitment, placement, and performance assessment that underpins the core of its affiliations in human resources and workforce management in the commercial and governmental arenas (Bloom, Krathwohl, 1956; Dave, 1970; Harrow, 1972; Krathwohl, Bloom, Bertram, 1973; Wiggins, McTighe, Ebrary, 2005).

Generally, the term [competence] refers to the performance standards associated with a profession or membership to a licensing organization. Assessing some level of performance in the workplace is frequently used as a competence measure, which means measuring aspects of the job at which a person is competent (IT2017, p. 28).

The meaning of *competency* may vary widely among particular professions or registries; IT2017 adopts its own "working" definition: *Competency = Knowledge + Skills + Dispositions*. In IT2017's "working" definition, *knowledge* is understood as mastery and the transfer of content knowledge. *Skills* are understood as capabilities and strategies for higher-order thinking and interactions with others and the world around. *Dispositions* are understood as personal qualities (socio-emotional skills, behaviors, attitudes) associated with success in higher education and career (IT2017, p. 28).

Competency's epistemological roots are found in the formal training of established labor disciplines (e.g. nursing) where the procedures and behavior employed require consistent, predictable, and disciplined application or treatments (Heath, 1998; Johns, 1995). They also align with the rubrics of socially acceptable conduct that circumscribe a specific profession with consequent statutory implications (e.g. licensure and legal liability). IT2017's citation of "six facets of learning transfer" intimate a predilection for a vocational trajectory of learning (IT2017, p. 28; Wiggins, 2005). (See Table 1.)

Curricular descriptions based upon *competency* aspire to "close the loop" on experiential learning, exploit *learning by doing*, and systematize performance-based assessment. At the same time, competencies of themselves are indifferent to specific pedagogy. In that sense, competency focuses on the accomplished learner's professional capacity rather than the agency of pedagogy.

5. MODELING CURRICULUM AS COMPETENCIES

In the simplest terms, curriculum developers adopt competencies to model the "product" of an educational process. Given a set of known facts defining a domain of interest, the function of competency-based education is envisioned as imbuing an understanding of practice in a subset of domain knowledge that suffices a desired range of practice; thus, supporting satisfactory performance in a particular profession, discipline, or situated task (Anderson, 2001). Curriculum is a model intended to define and instill competency.

In the following set theoretic representation, *Competency-Disposition-Knowledge-Skills-Task* (CDKST), we adopt three grounding propositions to conceptualize curriculum: 1) learning is acquiring knowledge elements arranged taxonomically that enable satisfactorily performing relevant tasks; 2) the concept of "skill" is a degree of mastery of a knowledge element modulated by disposition to achieve a valued outcome, and 3) disposition denotes the values and motivation that guide applying knowledge while designating the quality of knowing commensurate with a standard of professional performance.

Note: The original idea for this set theoretic model emerged in the collaborative work of the CC2020 "tool" Task Group of which the first author is a member. The collaborative process and contributions of CC2020 team as well as the "EDSIG Tool Auxiliary" are greatly appreciated.

CDKST Curriculum Framework

Competency-Disposition-Knowledge-Skills-Task

Knowledge elements, **K**, are factual concepts supported by science and/or professional practice that underpin a vocabulary of objects, behaviors, and relationships as the domain of interest in a discourse (be it curriculum, task, job, or profession). **S**, the skill attribute, denotes the *quality of knowing* (e.g. mastery, expertise, adeptness, or proficiency) that an accomplished learner must possess to satisfactorily apply a knowledge element in a circumstance of performance. In this sense it is the capacity to demonstrate a degree of *cognitive command* of that knowledge. In this conceptualization cognitive command is represented by Bloom's (revised) taxonomy of learning objectives: remember, understand, apply, analyze, evaluate, and create (See Appendix D, Anderson, 2001). Disposition, **D**, represents a commitment, motivation, toward an aspect of professional practice that reflects the attitude deemed critical to satisfaction in a professional circumstance or context. Task, **T**, is a situated instance of engaging knowledge with a degree of mastery. **C**, competency is a demonstrated sufficiency in a task with an appropriate disposition.

T = task
T --> $\{(K_i,S_j)\}$ knowledge applied

[A task is a set of one or more knowledge/skill pairs engaged in a purposeful act.]

Task, **T**, is *knowledge applied* in a "live" context to accomplish a designated purpose. **T** represents a *specification* of capability that curriculum is obligated to inculcate in the accomplished learner.

A task is the application of specific knowledge to a situation at hand. Note that tasks may be of varying complexity in terms of the range of knowledge elements engaged. Individual knowledge elements may participate in a variety of tasks. A task may be a collection of constituent tasks within which each knowledge element is applied with a distinct skill. As a collective, the task's satisfactory accomplishment exhibits a sufficiency of knowing and doing.

C = competency
C --> $\{(\Sigma(K_i,S_j) \mid (K_i,S_j) \in T), D_k\}$

[Competency is a task(s) satisfied demonstrating a disposition to professionalism.]

Competency, **C**, is the capacity to accomplish a task by applying knowledge and skills within a particular disposition. This is the goal sought by a competency-based perspective on curricular design. This forms a focus for assessment as each competency represents both a requirement and the instrument of certification to assure the learner's successful performance – success denoted by the satisfactory outcome of applying the knowledge modulated by the disposition. It is reasonable to expect that a system of competency specifications would form a telescopic or hierarchical arrangement of modularized task complexity and thus, would lead to an incremental or progressive process of learning and experience accumulation that would subsequently justify advancement to more elaborate, intricate, or difficult tasks or higher degrees of professionalism.

E = education
E --> $\{C_i\}$
B = baccalaureate degree
$\mathbf{B}_e$ --> $\{\sum(C_i) \mid C_i \in E\}$

[A baccalaureate is the cumulation of competencies comprising a course of learning.]

E, is a composition of competencies relevant to (or defining) a professional or academic course of study, a curriculum. A baccalaureate degree, **B**, is granted by an authorized institution. In fact, the list of competencies may be the vary testimony to the focus of an intended career direction shaping an academic program's intension. This would be the construct for comparing educational programs, assessing guideline or accreditation compliance, or prototyping distinct perspectives on the larger domain of knowledge such as across subdomains of *computing*!

J = job description
J --> $\{C_i\}$
JP = job permit
$\mathbf{JP}_j$ --> $\{\sum(C_i) \mid C_i \in J\}$

[A job permit is the cumulation of competencies comprising a job description.]

In its own fashion, a particular job description is in effect a "mini-curriculum" as it prescribes performance requirements that usually distinguish the desired applicant or employee attributes. The particulars of the organization, the industry, or the marketplace would shape both the collection of knowledge elements, skills, and the disposition of their application, thus, aligning with a particular vocation.

P = profession
P --> $\{J_i\}$
L = professional license
$\mathbf{L}_p$ --> $\{\sum(J_i) \mid J_i \in P\}$

[A professional license is the cumulation of competencies denoting the profession as a set of jobs that distinguish it.]

In this last aggregation, professional societies and governmental agencies specify collections of competencies that qualify a legal standing as a licensed professional (e.g. professional engineer, medical doctor, physician's assistant, nurse, a member of the bar, barber, cosmetologist, etc.).

The CDKST model does not attempt to shape or bound the dimensions of pedagogy as that requires integration with the cultural context within which it must be applied. However, pedagogy must align with the designated disposition modulating the *professionalism* the student must demonstrate as competency in context.

6. VISUALIZING THE CDKST FRAMEWORK

Degree program, job description, and certification requirements, all represent conceptually identical competency-based structures. They would differ primarily in the "footprint" of **K** and **S** in the universe that delineates their individually relevant domain of discourse. Figure 3 depicts the various CDKST modeling elements comprising three instances of competency collection. We can readily assume that the availability of a directory of knowledge elements provides a ready resource for sharing relevant understanding across subdisciplines. Somewhat "bottom up" this might be the foundation for curriculum designers to frame a curriculum and then, flesh it out with selected pedagogy as an educational program. Similarly, somewhat "top down" employers and accrediting agencies might use a directory of competencies to assess the similarity or difference among job descriptions or professions in evaluating workforce policies or human resource arrangements.

It is likely that students will navigate their choice of programs by first examining their options "top down" and refining their choice of specific educational program at least in part "bottom up."

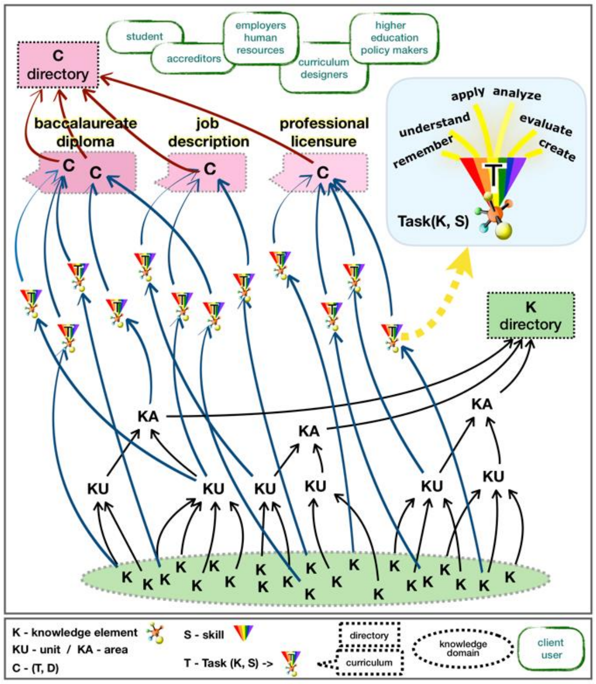


Figure 3 – CDKST Curriculum Framework

7. CHALLENGING CDKST DESIGN IMPLICATIONS

Indeed, by addressing some design implications in this section, we admit that our conceptualization is developmental rather than a complete design for the curriculum "tool." As the IT2017 Curriculum Guideline is the only published artifact currently available that is predominantly competency-focused, this section leans on that IT2017 document to discuss some challenging implications entailed by the set theoretic framework conceptualized above, CDKST.

Taxonomy – The first challenge for the CDKST framework is our first proposition that there must be a foundational taxonomy of knowledge elements of computing as in Figure 3 providing a basis to demonstrate any particular of competency.

If knowledge elements or competencies are to be comparable across curriculum descriptions, there must be a consistently shared curricular vocabulary. While both competency statements and knowledge elements are susceptible to revision in terminology and meaning, it seems clear that knowledge elements must be more stable, *less* likely to be deleted and replaced – more likely to be added to or fall into disuse. Competencies more so reflect a current policy or practitioner behavior often specific to an application context. These would seem far more dynamic than knowledge taxonomy. Epistemologically, competency is derived from knowledge, skills, and disposition rather than, knowledge, skills, and disposition derived from competency – these concepts are not commutatively derivative.

Paraphrasing the wisdom of Per Brinch Hansen – the [COBOL, Fortran, Algol, …] specification doesn't define the programming language, the compiler does (Hansen, 1973)! And so, competency does not inform the knowledge-skills-disposition but rather, knowledge-skills-disposition informs the competency.

Given infinite resources, the simplest plan for developing a *universal* taxonomy of computing knowledge might be to develop the taxonomy and then reconstruct all the existing curricular guidelines of computing. Even with infinite resources, the scope of such an effort is somewhat mind boggling. However, this taxonomy aspect of the conceptualization is basically unavoidable. The only apparent recourse is to protract the process of taxonomy development as an undertaking of "*continuous taxonomy improvement*."

The most likely "work around" for this task is extensive automated text analysis as a means of reducing the necessary manual effort by statistically detecting recurring language patterns to nominate prospective knowledge elements. The mining of text specific to *performance statements* in IT2017 and to KA-KU-LO statements in the existing guidelines along with course descriptions suggest the best opportunity for cataloging knowledge elements. (See Appendix A as an example of computing knowledge categorization and Appendix G that proposes performance statements applying IT skills.)

Correspondence of Knowledge Elements with Competency – Reviewing IT2017's current representation of the relationship between competency and performance statements, a rubric is needed to normalize the form and semantics for specifying competency that clearly expose the necessary constituent knowledge elements and dispositions.

IT2017 suggests action verbs to characterize task performance statements. (See Appendix C.) Most of its competencies are more specific in their reference to "outcomes" than to particular antecedent knowledge or skills that are necessary to satisfactory performance. Again, the most likely "work around" for this situation may be automated text analysis to identify prospective correspondences, "outcome *relies upon* knowledge element(s) applied with this skill(s)," to reduce the manual workload of transliterating competencies. (See Appendix F.) At this juncture of our design analysis the cognitive process dimension verbs of Bloom's (revised) in Appendix D appear most appropriate for characterizing the degree of knowledge mastery for satisfactorily applying required knowledge – alternatives may arise!

Dimensional characteristics – Although knowledge elements may have designated titles, their applications in distinct task instances will reflect different ways of "understanding" the element. These differences are critical in specifying, evaluating and, particularly in the "tool," visualizing competency.

A benefit of adopting the revised Bloom's approach to cognitive processes to differentiate the skill required in applying knowledge is that the levels are intrinsically cumulative – that is the higher levels require skill support at all the lower levels (Anderson, 2001). This characteristic is analogous to magnitude enabling numerous visualization possibilities.

In Figure 2 the CC2005 "footprint" visualization ascribes the y-axis to the "continuum" of hardware through organizational policy. This particular taxonomical characterization is congruous in concept to the *semiotic ladder* (Stamper, 1991). With a formidable grounding in theory, the semiotic ladder likewise distributes and delineates the communicating media of meaning as agency extending from the *material* at the base through the *conceptual* at the top. (See Table 2.) The semiotic ladder characterizes a progression where steps offer homologous arrays of structural and behavioral metaphors with which to express successive degrees of abstraction from the material to the organizational (Stamper, 1973; Liu, 2000).

Table 2 - The Semiotic Framework

Semiotic Ladder	Semiotic Layer Description
Social World	Beliefs, expectations, functions, commitments, contracts, law, culture
Pragmatics	Intensions, communications, conversations, negotiations
Semantics	Meanings, propositions, validity, truth, signification, denotations
Syntactics	Formal structure, language, logic, data, records, deduction, software, files
Empirics	Pattern, variety, noise, entropy, channel capacity, redundancy, efficiency, codes
Physical	Signals, traces, physical distinctions, hardware, component density, speed, economics

The x-axis in Figure 2 traces the application of computing knowledge from the theoretical on the left to the applied on the right. In our conception of that x-axis we propose to use the *quality of knowing* the knowledge element to intimate the degree of conceptual insight, capacity for judgement, appropriate for applying that knowledge satisfactorily. (See Figure 4.) The x-axis positioning from left to right on a continuum would range from basic recollection and understanding over to the conceptual insight needed for in depth evaluation and creativity following Bloom's revised cognitive dimensions of learning in Appendix D (Anderson, 2001).

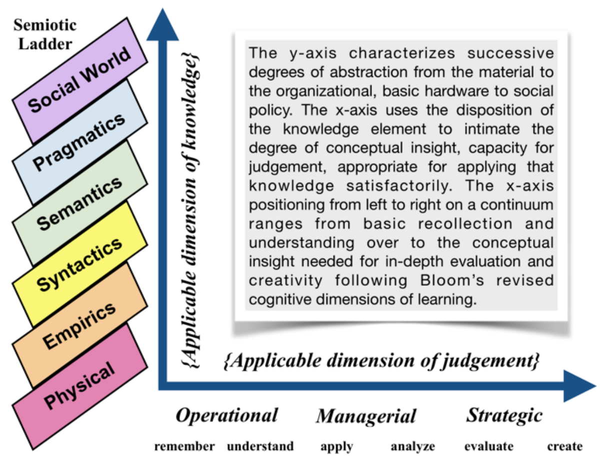


Figure 4 – Prospective CC2020 "Footprint"

Application Domain Competency – Naturally our attention is drawn primarily to knowledge elements in the domain of "computing." However, most computing curricula are embedded, intertwined, with the application domain to which they are applied. Curriculum in information systems is a particularly apt example embedded in commerce (Topi, 2017a). Also, virtually every aspect of computing relies to some degree on mathematics, verbal and written language, and organizational awareness – all of which are deserving of explicit education and thus, their own knowledge elements.

It may be possible to compartmentalize the knowledge of computing that is appropriate for information systems development in general. However, specific domain knowledge of an industry, regulatory, or cultural locale will probably require assessment of disposition adjustments to account for various degrees of indigenous sensitivity and relevance. This may lead to the merging or integrating curricular domains of computing with application specific competency. However, this is not a conceptual hurdle since the CDKST model (although targeted in this discussion to computing) appears to be commensurately applicable in any knowledge domain.

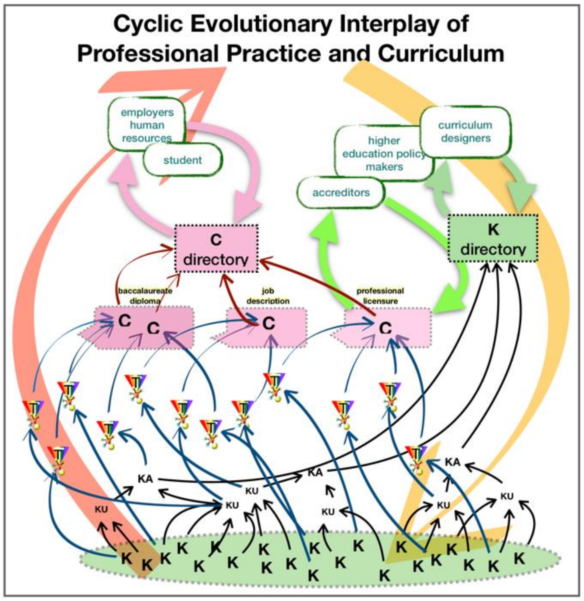


Figure 5 – Cyclic Evolution of Curricula

Reckoning with Competency Instability – Although the occurrence of computing curriculum retrospectives would seem to be infrequent (e.g. CC2005 – CC2020), the breadth and depth of computing's burgeoning influence on society far outpaces our current capacity for "self-reflection." To steward the investment in a competency-based taxonomy of computing knowledge proposed in this discourse, an ongoing curation program must be mounted not only to support the beneficial use of existing competency-based computing knowledge but also, the renewal and addition of new science, practice, and applications going forward. Although periodic publications chronicle the "contemporary state" of curricular evolution, that evolution is non-stop in breadth and depth. If the utility of the "tool" even weakly approximates the goals envisioned for examining, comparing, and developing curricular specifications, the "tool's" utility in academia may likely be eclipsed by industry interests in professional development, human resources, and by governmental policy makers involved in workforce analysis and development. In Figure 5, CDKST intimates four or more cycles of interactive co-evolution coupling computing practice with curriculum in response to the emerging science, technology, and applications of computing. While cycles provoked by science may span generations, technology applied often sparks rapid cycles best served by frequently reviewing and renewing various computing curricula.

8. Discussion

The CC2020 visualization tool must rely upon an interoperability and interrelationships encompassing competency, knowledge, skills, and disposition as spatially oriented concepts. In this discourse we presented a conceptualized design that exposes the epistemological and computational challenges that the "tool" must surmount to achieve the goals set out for it in CC2020.

The initiative to advance CC2005's conception of the visual representation of curriculum from the state of an interpretive sketch to a computationally accurate representation of curriculum specification is significant in multiple dimensions:

a) The goal of an empirically accurate representation of curricular specifications advances a normalized definition of computing and its components to correlate the perspectives of all the sub-disciplines of computing.
b) The ability to analyze and prototype curricular specifications through visual manipulation encourages a more active participation in curriculum development involving industry, government, and the public.
c) The "tool" offers the prospect of significantly increasing transparency of computing curricula to the benefit of students, academic programs, and employers. The "tool" can advertise the metaphorical "list of ingredients" in a more understandable and digestible form for curriculum consumers.
d) Technological, theoretical, and professional practice developments that naturally emerge in subdisciplines can be identified and explored earlier for their co-relevancy and implications across subdisciplines. This should facilitate opportunities for enhanced cooperation among interested parties and enable tangible economies of effort for researchers, funding agencies, and academic institutions.
e) An ongoing "continuous taxonomy improvement" process at the intersection of subdiscipline planning, development, and maintenance provides an opportunity for synchronizing and streamlining the collaborative efforts of curriculum developers, professional societies, and program assessment / accreditation agencies.

f) The prospect of compatible representations of job and curricular competencies can offer a great incentive for industry, government, and academia collaboration.
g) The potential exists for better understanding and exploiting various modes of learning and the relationships among computing education programs: primary, secondary, post-secondary, baccalaureate, graduate, certification, continuing professional development, etc. – perhaps, even a better understanding of the computing discipline as a whole.

We admit that the vision of a *universal* knowledge taxonomy of computing is unreasonable within the time horizon of the CC2020 project. However, a proof of concept and phased launch are doable. In the end, a crowd-sourced, "continuous taxonomy improvement" effort would be an invaluable legacy of the CC2020 project. We can hope that the nascent foundation provided by the "tool" will attract academics, professional societies, and workforce specialists who will contribute their own ideas and economic support as investment in the future of computing education for generations to come.

9. ACKNOWLEDGEMENTS

The ideas presented in this paper derive from a broad spectrum of experiences over the past forty or so years of the authors' computing careers: proposing, implementing, reviewing, and remodeling computing curricula. But, the impetus leading to the convergence of the thoughts presented herein is largely attributable to the collaborative energies and shared commitment of the CC2020 "tool" team. Although the authors accept full liability for the presentation herein, we wish to recognize and express our gratitude for the invaluable sharing of experience and imaginative ideas that emerged among the "tool" team members: Shingo Takada, Ernesto Cuadros-Vargas, Gerritt van der Veer, Steven Gordon, Linda Marshall, and Tania McVeety. We also recognize our EDSIG colleagues Musa Jafar and Jason Sharp who have contributed to the CC2020 "tool" project as members of the "EDSIG Tool Auxiliary." There is much work ahead of the "tool" team!

10. REFERENCES

Anderson, L. W., and D.R. Krathwohl (2001). *A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of educational objectives*. Longman, New York, USA.

Barr, R.B. and Tagg, J. (1995). "From Teaching to Learning: A New Paradigm for Undergraduate Education." *Change*, 27(5), 12-25.

Bloom, B.S. and Krathwohl, D. R. (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals, by a committee of college and university examiners. Handbook I: Cognitive Domain*. Longmans, Green, New York, USA.

Coursera, (2018). https://about.coursera.org, (current July 4, 2018).

Dave, R.H. (1970). *Psychomotor levels in Developing and Writing Behavioral Objectives*, pp.20-21. R.J. Armstrong, ed. Tucson, Arizona: Educational Innovators Press.

Hansen, P.B. (1973). *Operating Systems* Principles, Prentice-Hall, Upper Saddle River, NJ, USA.

Harrow, A. (1972). *A Taxonomy of Psychomotor Domain: A Guide for Developing Behavioral Objectives*. David McKay Co., Inc., New York, USA.

Heath, H. (1998). "Reflection and patterns of knowing in nursing," *Journal of Advanced Nursing*, Vol. 27, pp. 1054-1059.

IT2017JointTaskGroup (2017). *Information Technology Curricula 2017*. ACM and IEEE-CS, https://www.acm.org/education/curricula-recommendations (current July 4, 2018).

Johns, C. (1995). "Framing learning through reflection within Carper's fundamental ways of knowing in nursing," *Journal of Advanced Nursing,* Vol. 22, pp. 226-234.

Krathwohl, D. R., Bloom, B. S., & Bertram, B. M. (1973). *Taxonomy of Educational Objectives, the Classification of Educational Goals. Handbook II: Affective Domain*. David McKay Co., Inc., New York, USA.

Liu, K. (2000). *Semiotics in Information Systems Engineering*, Cambridge University Press, Cambridge, UK.

Longenecker, H. E., Feinstein, D. L., & Babb, J. S. (2013). Is there a need for a Computer Information Systems model curriculum? In Proceedings of the Information Systems Educators Conference, ISSN: 2167, 1435-1447.

Shackelford, R., McGettrick, A., Sloan, R., Topi, H., Davies, G., Kamali, R., Cross, J., Impagliazzo, J., LeBlanc, R., & Lunt, B. (2005). *Computing Curricula 2005: The Overview Report*. ACM, AIS, and IEEE-CS. https://www.acm.org/education/curricula-recommendations (current July 4, 2018).

Stamper, R. K. (1973). *Information in Business and Administrative Systems*, John Wiley and Sons, New York, NY.

Stamper, R. K. (1991). "The semiotic framework for information systems research. Information systems research," *Contemporary approaches and emergent traditions*, 515-528.

Topi, H. (2017). "Role of Information Systems in the CC2020 Initiative". *ACM Inroads*, 8(4).

Topi, H, (2017). "Information Systems in CC2020: Comparing Key Structural Elements of Curriculum Recommendations in Computing". *2017 Proceedings*. 9. http://aisel.aisnet.org/siged2017/9

USDoE (2018). "Competency-Based Learning or Personalized Learning," U.S. Department of Education, https://www.ed.gov/oii-news/competency-based-learning-or-personalized-learning (current July 4, 2018)

Waguespack, L., Babb, J. S., Yates, D. (2018). "Triangulating Coding Bootcamps in IS Education: Bootleg Education or Disruptive Innovation?" *Information Systems Education Journal,* 16(6) pp 48-58. http://isedj.org/2018-16/ ISSN: 1545-679X.

Wiggins, G., McTighe, J., and Ebrary, I. (2005). *Understanding by design* (Expanded second edition). Alexandria, VA: Association for Supervision and Curriculum Development.

Wiggins, G., and McTighe, J. 2011. *The Understanding by Design Guide to Creating High-Quality Units*. Alexandria, VA: Association for Supervision and Curriculum Development.

Appendix A – Computing Topics Weighting in CC2005

Knowledge Area	CE		CS		IS		IT		SE	
	min	max	min	max	min	max	min	max	min	max
Programming Fundamentals	4	4	4	5	2	4	2	4	5	5
Integrative Programming	0	2	1	3	2	4	3	5	1	3
Algorithms and Complexity	2	4	4	5	1	2	1	2	3	4
Computer Architecture and Organization	5	5	2	4	1	2	1	2	2	4
Operating Systems Principles & Design	2	5	3	5	1	1	1	2	3	4
Operating Systems Configuration & Use	2	3	2	4	2	3	3	5	2	4
Net Centric Principles and Design	1	3	2	4	1	3	3	4	2	4
Net Centric Use and configuration	1	2	2	3	2	4	4	5	2	3
Platform technologies	0	1	0	2	1	3	2	4	0	3
Theory of Programming Languages	1	2	3	5	0	1	0	1	2	4
Human-Computer Interaction	2	5	2	4	2	5	4	5	3	5
Graphics and Visualization	1	3	1	5	1	1	0	1	1	3
Intelligent Systems (AI)	1	3	2	5	1	1	0	0	0	0
Information Management (DB) Theory	1	3	2	5	1	3	1	1	2	5
Information Management (DB) Practice	1	2	1	4	4	5	3	4	1	4
Scientific computing (Numerical mthds)	0	2	0	5	0	0	0	0	0	0
Legal / Professional / Ethics / Society	2	5	2	4	2	5	2	4	2	5
Information Systems Development	0	2	0	2	5	5	1	3	2	4
Analysis of Business Requirements	0	1	0	1	5	5	1	2	1	3
E-business	0	0	0	0	4	5	1	2	0	3
Analysis of Technical Requirements	2	5	2	4	2	4	3	5	3	5
Engineering Foundations for SW	1	2	1	2	1	1	0	0	2	5
Engineering Economics for SW	1	3	0	1	1	2	0	1	2	3
Software Modeling and Analysis	1	3	2	3	3	3	1	3	4	5
Software Design	2	4	3	5	1	3	1	2	5	5
Software Verification and Validation	1	3	1	2	1	2	1	2	4	5
Software Evolution (maintenance)	1	3	1	1	1	2	1	2	2	4
Software Process	1	1	1	2	1	2	1	1	2	5
Software Quality	1	2	1	2	1	2	1	2	2	4
Comp Systems Engineering	5	5	1	2	0	0	0	0	2	3
Digital logic	5	5	2	3	1	1	1	1	0	3
Embedded Systems	2	5	0	3	0	0	0	1	0	4
Distributed Systems	3	5	1	3	2	4	1	3	2	4
Security: issues and principles	2	3	1	4	2	3	1	3	1	3
Security: implementation and mgt	1	2	1	3	1	3	3	5	1	3
Systems administration	1	2	1	1	1	3	3	5	1	2
Management of Info Systems Org.	0	0	0	0	3	5	0	0	0	0
Systems integration	1	4	1	2	1	4	4	5	1	4
Digital media development	0	2	0	1	1	2	3	5	0	1
Technical support	0	1	0	1	1	3	5	5	0	1

The min value represents the minimum emphasis typically applied to the topic based upon the discipline-specific curriculum guidelines. The max value represents the greatest emphasis (Shackelford, 2005, p. 24).

Appendix B – Comparison of Computing Curricula

	CE2016	CS2013	IS2010	IT2017	MSIS2016	SE2014
Core structure	Knowledge area – Knowledge unit – Learning outcome	Knowledge area – Knowledge unit – Topic/ Learning outcome	Course – Topic/ Learning outcome	IT Domain – Competency	Competency area – Competency category – Competency	Knowledge area – Knowledge unit – Topic
Body of knowledge	Core structure	Core structure	Supporting material	No	No	Core structure
Detailed learning outcomes	Associated with Knowledge units	Associated with Knowledge units	Associated with Courses	"Performances" associated with subdomains are closely related	No	No
High-level graduate characteristics	Yes	Yes	Yes	Yes	Yes	Yes
Competency framework	No	No	No	Yes	Yes	No
Professional profiles	No	No	Yes	No	Yes	No
Note: The original idea for this table emerged from collaborative work at the August 2017 CC2020 Task Force meeting. The collaborative process and contributions of CC2020 are greatly appreciated.						

Adapted from (Topi, 2017b)

Appendix C – Bloom's (Revised) of Educational Objectives: Performance Verbs Proposed in IT2017

Explain	Interpret	Apply	Demonstrate Perspective	Show Empathy	Have Self-Knowledge
demonstrate derive describe how design exhibit express induce instruct justify model predict prove show how synthesize teach	create analogies critique document evaluate illustrate judge make sense of make meaning of provide metaphors read between the lines represent tell a story of translate	adapt build create debug decide design exhibit invent perform produce propose solve test use	analyze argue compare contrast criticize infer	assume role of be like be open to believe consider imagine relate role play	be aware of realize recognize reflect self-assess

IT2017, Table 6.4: Performance verbs to generate ideas for performance goals and professional practice [Wiggins, McTighe, 2011]

Appendix D – Bloom's (Revised) of Educational Objectives: Facets of Learning (Anderson, 2001)

Appendix E - IT2017 Domains of Competency

Essential IT Domains and Levels of Student Engagement	
ITE-CSP Cybersecurity Principles [6%] ITE-CSP-01 Perspectives and impact [L1] ITE-CSP-02 Policy goals and mechanisms [L1] ITE-CSP-03 Security services, mechanisms, and countermeasures [L2] ITE-CSP-04 Cyber-attacks and detection [L2] ITE-CSP-05 High assurance systems [L2] ITE-CSP-06 Vulnerabilities, threats, and risk [L2] ITE-CSP-07 Anonymity systems [L1] ITE-CSP-08 Usable security [L1] ITE-CSP-09 Cryptography overview [L1] ITE-CSP-10 Malware fundamentals [L1] ITE-CSP-11 Mitigation and recovery [L1] ITE-CSP-12 Personal information [L1] ITE-CSP-13 Operational issues [L2] ITE-CSP-14 Reporting requirements [L1]	**ITE-GPP Global Professional Practice** [3%] ITE-GPP-01 Perspectives and impact [L1] ITE-GPP-02 Professional issues and responsibilities [L1] ITE-GPP-03 IT governance and resource management [L1] ITE-GPP-04 Risk identification and evaluation [L1] ITE-GPP-05 Environmental issues [L1] ITE-GPP-06 Ethical, legal, and privacy issues [L1] ITE-GPP-07 Intellectual property [L1] ITE-GPP-08 Project management principles [L1] ITE-GPP-09 Communications [L1] ITE-GPP-10 Teamwork and conflict management [L1] ITE-GPP-11 Employability skills and careers in IT [L1] ITE-GPP-12 Information systems principles [L1]
ITE-IMA Information Management [6%] ITE-IMA-01 Perspectives and impact [L1] ITE-IMA-02 Data-information concepts [L2] ITE-IMA-03 Data modeling [L3] ITE-IMA-04 Database query languages [L3] ITE-IMA-05 Data organization architecture [L3] ITE-IMA-06 Special-purpose databases [L1] ITE-IMA-07 Managing the database environment [L2]	**ITE-IST Integrated Systems Technology** [3%] ITE-IST-01 Perspectives and impact [L1] ITE-IST-02 Data mapping and exchange [L2] ITE-IST-03 Intersystem communication protocols [L2] ITE-IST-04 Integrative programming [L2] ITE-IST-05 Scripting techniques [L2] ITE-IST-06 Defensible integration [L1]
ITE-NET Networking [5%] ITE-NET-01 Perspectives and impact [L1] ITE-NET-02 Foundations of networking [L1] ITE-NET-03 Physical layer [L2] ITE-NET-04 Networking and interconnectivity [L3] ITE-NET-05 Routing, switching, and internetworking [L2] ITE-NET-06 Application networking services [L2] ITE-NET-07 Network management [L3]	**ITE-PFT Platform Technologies** [1%] ITE-PFT-01 Perspectives and impact [L1] ITE-PFT-02 Operating systems [L3] ITE-PFT-03 Computing infrastructures [L1] ITE-PFT-04 Architecture and organization [L1] ITE-PFT-05 Application execution environment [L1]
ITE-SPA System Paradigms [6%] ITE-SPA-01 Perspectives and impact [L1] ITE-SPA-02 Requirements [L2] ITE-SPA-03 System architecture [L1] ITE-SPA-04 Acquisition and sourcing [L2] ITE-SPA-05 Testing and quality assurance [L2] ITE-SPA-06 Integration and deployment [L2] ITE-SPA-07 System governance [L2] ITE-SPA-08 Operational activities [L3] ITE-SPA-09 Operational domains[L3] ITE-SPA-10 Performance analysis [L1]	**ITE-SWF Software Fundamentals** [4%] ITE-SWF-01 Perspectives and impact [L1] ITE-SWF-02 Concepts and techniques [L2] ITE-SWF-03 Problem-solving strategies [L1] ITE-SWF-04 Program development [L3] ITE-SWF-05 Fundamental data structures [L2] ITE-SWF-06 Algorithm principles and development [L2] ITE-SWF-07 Modern app programming practices [L1]
ITE-UXD User Experience Design [3%] ITE-UXD-01 Perspectives and impact [L1] ITE-UXD-02 Human factors in design [L2] ITE-UXD-03 Effective interfaces [L2] ITE-UXD-04 Application domain aspects [L1] ITE-UXD-05 Affective user experiences [L1] ITE-UXD-06 Human-centered evaluation [L1] ITE-UXD-07 Assistive technologies and accessibility [L1] ITE-UXD-08 User advocacy [L1]	**ITE-WMS Web and Mobile Systems** [3%] ITE-WMS-01 Perspectives and impact [L1] ITE-WMS-02 Technologies [L2] ITE-WMS-03 Digital media [L2] ITE-WMS-04 Applications concepts [L2] ITE-WMS-05 Development Frameworks [L2] ITE-WMS-06 Vulnerabilities [L1] ITE-WMS-07 Social software [L1]

Essential IT Domains (IT2017, Table 6.2a, p. 50)

Appendix E - IT2017 Domains of Competency (continued)

Supplemental IT Domains and Levels of Student Engagement	
ITS-ANE	**Applied Networks** [4%]
ITS-ANE-01	Proprietary networks [L2]
ITS-ANE-02	Network programming [L2]
ITS-ANE-03	Routing protocols [L2]
ITS-ANE-04	Mobile networks [L2]
ITS-ANE-05	Wireless networks [L2]
ITS-ANE-06	Storage area networks [L1]
ITS-ANE-07	Applications for networks [L2]
ITS-CCO	**Cloud Computing** [4%]
ITS-CCO-01	Perspectives and impact [L1]
ITS-CCO-02	Concepts and fundamentals [L2]
ITS-CCO-03	Security and data considerations [L2]
ITS-CCO-04	Using cloud computing applications [L2]
ITS-CCO-05	Architecture [L2]
ITS-CCO-06	Development in the cloud [L2]
ITS-CCO-07	Cloud infrastructure and data [L2]
ITS-CEC	**Cybersecurity Emerging Challenges** [4%]
ITS-CEC-01	Case studies and lessons learned [L1]
ITS-CEC-02	Network forensics [L2]
ITS-CEC-03	Stored data forensics [L2]
ITS-CEC-04	Mobile forensics [L1]
ITS-CEC-05	Cloud security [L1]
ITS-CEC-06	Security metrics [L1]
ITS-CEC-07	Malware analysis [L1]
ITS-CEC-08	Supply chain and software assurance [L1]
ITS-CEC-09	Personnel and human security [L1]
ITS-CEC-10	Social dimensions [L1]
ITS-CEC-11	Security implementations [L1]
ITS-CEC-12	Cyber-physical systems and the IoT [L1]
ITS-DSA	**Data Scalability and Analytics** [4%]
ITS-DSA-01	Perspectives and impact [L1]
ITS-DSA-02	Large-scale data challenges [L2]
ITS-DSA-03	Data management [L2]
ITS-DSA-04	Methods, techniques, and tools [L2]
ITS-DSA-05	Data governance [L2]
ITS-DSA-06	Applications [L2]
ITS-IOT	**Internet of Things** [4%]
ITS-IOT-01	Perspectives and impact [L1]
ITS-IOT-02	IoT architectures [L2]
ITS-IOT-03	Sensor and actuator interfacing [L1]
ITS-IOT-04	Data acquisition [L1]
ITS-IOT-05	Wireless sensor networks [L2]
ITS-IOT-06	Ad-hoc networks [L1]
ITS-IOT-07	Automatic control [L2]
ITS-IOT-08	Intelligent information processing [L2]
ITS-IOT-09	IoT application and design [L2]
ITS-MAP	**Mobile Applications** [3%]
ITS-MAP-01	Perspectives and impact [L1]
ITS-MAP-02	Architectures [L1]
ITS-MAP-03	Multiplatform mobile application development [L2]
ITS-MAP-04	Servers and notifications [L1]
ITS-MAP-05	Performance issues [L1]
ITS-MAP-06	Views and gestures [L1]
ITS-MAP-07	Interface implementations [L2]
ITS-MAP-08	Camera, state, and documents interaction [L1]
ITS-MAP-09	2D graphic and animation [L1]
ITS-SDM	**Software Development and Management** [2%]
ITS-SDM-01	Process models and activities [L2]
ITS-SDM-02	Platform-based development [L1]
ITS-SDM-03	Tools and services [L2]
ITS-SDM-04	Management [L2]
ITS-SDM-05	Deployment, operations, maintenance [L2]
ITS-SRE	**Social Responsibility** [2%]
ITS-SRE-01	Social context of computing [L2]
ITS-SRE-02	Goals, plans, tasks, deadlines, and risks [L2]
ITS-SRE-03	Government role and regulations [L1]
ITS-SRE-04	Global challenges and approaches [L1]
ITS-SRE-05	Risk management [L1]
ITS-SRE-06	Sustainable Computing [L1]
ITS-VSS	**Virtual Systems and Services** [4%]
ITS-VSS-01	Perspectives and impact [L1]
ITS-VSS-02	Application of virtualization [L2]
ITS-VSS-03	User platform virtualization [L1]
ITS-VSS-04	Server virtualization [L1]
ITS-VSS-05	Network virtualization [L2]
ITS-VSS-06	Cluster design and administration [L2]
ITS-VSS-07	Software cluster applications [L2]
ITS-VSS-08	Storage [L1]

Supplemental IT Domains (IT2017, Table 6.2b, p. 51)

IT Essential Mathematics and Levels of Student Engagement	
ITM-DSC	**Discrete Structures**
ITM-DSC-01	Perspectives and impact [L1]
ITM-DSC-02	Sets [L1]
ITM-DSC-03	Functions and relations [L1]
ITM-DSC-04	Proof techniques [L1]
ITM-DSC-05	Logic [L1]
ITM-DSC-06	Boolean algebra principles [L1]
ITM-DSC-07	Minimization [L1]
ITM-DSC-08	Graphs and trees [L2]
ITM-DSC-09	Combinatorics [L1]
ITM-DSC-10	Iteration and recursion [L1]
ITM-DSC-11	Complexity Analysis [L1]
ITM-DSC-12	Discrete information technology applications [L1]

Related IT Essential Mathematics (IT2017, Table 6.2c, p. 52)

Appendix F - IT2017 Essential IT Domain Cluster Competency Examples

ITE-IMA Domain : Information Management	
Scope 1. Tools and techniques for efficient data modeling, collection, organization, retrieval, and management. 2. How to extract information from data to make data meaningful to the organization. 3. How to develop, deploy, manage and integrate data and information systems to support the organization. 4. Safety and security issues associated with data and information. 5. Tools and techniques for producing useful knowledge from information.	**Competencies** A. Express how the growth of the internet and demands for information have changed data handling and transactional and analytical processing, and led to the creation of special purpose databases. (*Requirements*) B. Design and implement a physical model based on appropriate organization rules for a given scenario including the impact of normalization and indexes. (*Requirements and development*) C. Create working SQL statements for simple and intermediate queries to create and modify data and database objects to store, manipulate and analyze enterprise data. (*Testing and performance*) D. Analyze ways data fragmentation, replication, and allocation affect database performance in an enterprise environment. (*Integration and evaluation*) E. Perform major database administration tasks such as create and manage database users, roles and privileges, backup, and restore database objects to ensure organizational efficiency, continuity, and information security. (*Testing and performance*)
Subdomains	
ITE-IMA-01 Perspectives and impact [L1]	ITE-IMA-05 Data organization architecture [L3]
ITE-IMA-02 Data-information concepts [L2]	ITE-IMA-06 Special-purpose databases [L1]
ITE-IMA-03 Data modeling [L3]	ITE-IMA-07 Managing the database environment [L2]
ITE-IMA-04 Database query languages [L3]	

ITE-IMA Domain: Information Management (IT2017, p. 56)

ITE-SWF Domain: Software Fundamentals	
Scope 1. Skills and fundamental programming concepts, data structures, and algorithmic processes 2. Programming strategies and practices for efficient problem solving 3. Programming paradigms to solve a variety of programming problems	**Competencies** A. Use multiple levels of abstraction and select appropriate data structures to create a new program that is socially relevant and requires teamwork. (*Program development*) B. Evaluate how to write a program in terms of program style, intended behavior on specific inputs, correctness of program components, and descriptions of program functionality. (*App development practices*) C. Develop algorithms to solve a computational problem and explain how programs implement algorithms in terms of instruction processing, program execution, and running processes. (*Algorithm development*) D. Collaborate in the creation of an interesting and relevant app (mobile or web) based on user experience design, functionality, and security analysis and build the app's program using standard libraries, unit testing tools, and collaborative version control. (*App development practices*)
Subdomains	
ITE-SWF-01 Perspectives and impact [L1]	ITE-SWF-05 Fundamental data structures [L2]
ITE-SWF-02 Concepts and techniques [L2]	ITE-SWF-06 Algorithm principles and development [L2]
ITE-SWF-03 Problem-solving strategies [L1]	ITE-SWF-07 Modern app programming practices [L1]
ITE-SWF-04 Program development [L3]	

ITE-SWF Domain: Software Fundamentals (IT2017, p. 58)

Appendix G - IT2017 IT Domain Performances

ITE-IMA Information Management (IT2017, p. 92)

ITE-IMA-01 Perspectives and impact

a. Describe how data storage and retrieval has changed over time.
b. Justify the advantages of a database approach compared to traditional file processing.
c. Describe how the growth of the internet and demands for information for users outside the organization (customers and suppliers) impact data handling and processing.
d. Tell a brief history of database models and their evolution.

ITE-IMA-02 Data-information concepts

a. Describe the role of data, information, and databases in organizations.
b. Compare and use key terms such as: information, data, database, database management system, metadata, and data mining.
c. Illustrate data quality, accuracy, and timeliness, and explain how their absence will impact organizations.
d. Describe mechanisms for data collection and their implications (automated data collection, input forms, sources).
e. Describe basic issues of data retention, including the need for retention, physical storage, backup, and security.

ITE-IMA-03 Data modeling

a. Design Entity Relationship diagrams based on appropriate organizational rules for a given scenario.
b. Describe the relationship between a logical model and a physical model.
c. Evaluate importance of database constraints.
d. Design a physical model for the best performance including impact of normalization and indexes.
e. Compare and contrast the differences and similarities between the relational and the dimensional data modeling (OLTP vs. OLAP).

ITE-IMA-04 Database query languages

a. Create, modify, and query database objects using the Structured Query Language (SQL).
b. Perform filtering and sorting data using various clauses including where, order by, between, like, group by, and having.
c. Use joins to select data across multiple tables.
d. Use embedded SQL queries.
e. Perform calculations in a query using calculated fields and aggregate functions.
f. Create updatable and non-updatable views.

ITE-IMA-05 Data organization architecture

a. Demonstrate select, project, union, intersection, set difference, and natural join relational operations using simple example relations provided.
b. Contrast and compare relational databases concepts and non-relational databases including object-oriented, XML, NewSQL and NoSQL databases.
c. Express the relationship between functional dependencies and keys, and give examples.
d. Evaluate data integrity and provide examples of entity and referential integrity.
e. Analyze how data fragmentation, replication and allocation affect database performance.

ITE-IMA-06 Special-purpose databases

a. Describe major concepts of object oriented, XML, NewSQL, and NoSQL databases.
b. Demonstrate an understanding of online analytical processing and data warehouse systems.
c. Describe methods of data mining and what insights may be gained by these methods.

ITE-IMA-07 Managing the database environment

a. Contrast and compare data administration and database administration.
b. Describe tasks commonly performed by database administrators.
c. Create and manage database users, roles, and privileges.
d. Consider the concept of database security and backup and recovery.
e. Evaluate the importance of metadata in database environment.

ITE-SWF Software Fundamentals *(IT2017, p. 96)*

ITE-SWF-01 Perspectives and impact

a. Reflect on how the creation of software has changed our lives.
b. Synthesize how software has helped people, organizations, and society to solve problems.
c. Describe several ways in which software has created new knowledge.

ITE-SWF-02 Concepts and techniques

a. Compare multiple levels of abstraction to write programs (constants, expressions, statements, procedures, parameterization, and libraries).
b. Select appropriate built-in data types and library data structures (abstract data types) to model, represent, and process program data.
c. Use procedures and parameterization to reduce the complexity of writing and maintaining programs and to generalize solutions.
d. Explain multiple levels of hardware architecture abstractions (processor, special purpose cards, memory organization, and storage) and software abstractions (source code, integrated components, running processes) involved in developing complex programs.
e. Create new programs by modifying and combining existing programs.

Appendix G - IT2017 IT Domain Performances (continued)

ITE-SWF-03 Problem-solving strategies

a. Explain abstractions used to represent digital data.
b. Develop abstractions when writing a program or an IT artifact.
c. Apply decomposition strategy to design a solution to a complex problem.
d. Explain appropriateness of iterative and recursive problem solutions.
e. Write programs that use iterative and recursive techniques to solve computational problems.

ITE-SWF-04 Program development

a. Develop a correct program to solve problems by using an iterative process, documentation of program components, and consultation with program users.
b. Use appropriate abstractions to facilitate writing programs: collections, procedures, application programming interfaces, and libraries.
c. Evaluate how a program is written in terms of program style, intended behavior on specific inputs, correctness of program components, and descriptions of program functionality.
d. Develop a program by using tools relevant to current industry practices: version control, project hosting, and deployment services.
e. Demonstrate collaboration strategies that consider multiple perspectives, diverse talents, and sociocultural experiences.

ITE-SWF-05 Fundamental data structures

a. Write programs that use data structures (built-in, library, and programmer-defined): strings, lists, and maps.
b. Analyze the performance of different implementations of data structures.
c. Decide on appropriate data structures for modeling a given problem.
d. Explain appropriateness of selected data structures.

ITE-SWF-06 Algorithm principles and development

a. Describe why and how algorithms solve computational problems.
b. Create algorithms to solve a computational problem.
c. Explain how programs implement algorithms in terms of instruction processing, program execution, and running processes.
d. Apply appropriate mathematical concepts in programing: expressions, abstract data types, recurrence relations, and formal reasoning on algorithm's efficiency and correctness.
e. Evaluate empirically the efficiency of an algorithm.

ITE-SWF-07 Modern app programming practices

a. Create web and mobile apps with effective interfaces that respond to events generated by rich user interactions, sensors, and other capabilities of the computing device.
b. Analyze usability, functionality, and suitability of an app program.
c. Collaborate in the creation of interesting and relevant apps.
d. Build and debug app programs using standard libraries, unit testing tools, and debuggers.
    1. Evaluate readability and clarity of app programs based on program style, documentation, pre- and post-conditions, and procedural abstractions.

Engaging College Students on Collaborative Projects with People with Cognitive Disabilities through e-Portfolios

James P. Lawler
lawlerj@aol.com

Anthony Joseph
ajoseph2@pace.edu

Pace University
163 William Street – 2nd Floor
New York City, New York 10038 USA

Abstract

College students can be advocates for people with disabilities. Collaboration on community projects in a school of computer science and information systems can be desirable for people with disabilities and students. The authors analyze the collaborative impacts from courses in Web Design for Non-Profit Organizations with moderately impaired but nimble people with cognitive disabilities. The authors find course features of e-Portfolios facilitating the engagement and the advocacy of the students for mentored people with disabilities. The findings can help instructors in information systems in cultivating e-Portfolios on course projects of students with those with disabilities.

Keywords: community engagement, digital portfolios, disabilities, e-Portfolios, information systems curricular, non-profit organizations, people with disabilities.

1. BACKGROUND

Digital portfolios or e-Portfolios are an amassment or "a collection of artifacts [containing] demonstrations [or examples] [from course projects] of individual [students of a university] (Lorenzo & Ittleson, 2005). e-Portfolios document evidence of the experiences of individual students (Garrison & Ring, 2013). Evidence of experiences can be documented in blogs, critiques and journals and in presentations and projects of information systems students, especially in interactions with others on new In fact, inclusion of e-Portfolio systems is increasing due to collaborative community technologies of the Web, which is highlighted in the literature (Amaya, Agudo, Sanchez, Rico & Hernandez-Linares, 2013). Importantly, millennial students in schools of information systems are increasingly involved with e-Portfolio projects, such as public service (Yancey, 2009). e-Portfolios can chronicle diverse experiences and document outcomes and reflections of the students (Light, Chen & Ittelson, 2012, pp.46,78). Literature considers e-Portfolios as essential for the increased learning of students of a university (Eynon & Gambino, 2017).

e-Portfolios are common in higher education (Dahlstrom, 2015). For example, half of private and public institutions in the country engage e-Portfolios for students (Clark & Eynon, 2009).

systems, due to the sociality of Web 2.0 tools (Exter, Rowe, Boyd & Lloyd, 2012). e-Portfolios as facilities are favored increasingly for housing the learning reflections of students of a university (Miller & Morgaine, 2009, p.2).

Essentially e-Portfolios can document the experiences of students and enable evaluations of their experiences. Features of blogs, critiques and journals and of presentations and project results can document learning from reflections of the students. Literature considers e-Portfolio functionality as helpful in increasing the learning outcomes from course projects of students (Laird, Shoup & Kuh, 2005). Given impacts of the learning on students indicated in the literature, a paper on projects in a school of computer science and information systems, involving more reflections and rich sources than on other projects, can be beneficial in learning more of the power of e-Portfolio systems in regards to community projects (Reynolds & Patton, 2014). Therefore, this paper is on community diversity projects of public service, involving the openness of students in partnership with a marginalized population of society: people with cognitive disabilities (Eynon & Gambino, 2017, p.42; Milsom, 2017).

2. INTRODUCTION TO PAPER

The courses consist of Web Design for Non-Profit Organizations projects, in the Seidenberg School of Computer Science and Information Systems of Pace University, engaging students in collaboratively designing Web sites for and with persons with cognitive disabilities. In engaging face-to-face with moderately impaired persons with disabilities on the projects, the students learn the needs of this population for entrepreneurial Web sites (Accardo & Whitman, 2011; Davis, 2015). The engagements are formed as individualized mentorships or partnerships of the persons with disabilities and the students on cooperative-learning production teams and are fruitful as the persons with disabilities are already nimble in co-creating Web sites with the students with non-programming simple Web template tools, such as www.wix.com (Salisbury, Gallucci, Palombaro & Peck, 1995; Eynon & Gambino, 2017, p.69). Experiences from the partnerships are denoted in documentation in blogs, critiques and journals of e-Portfolios by the students (Light, Chen & Ittelson, 2012, pp.87-88). The impacts of engagement and advocacy in learning about a misperceived neglected population of society are the desired learning outcomes of the course projects (Horowitz, Rawe & Whittaker, 2017; Wehmeyer, 2013).

The courses covered in the paper consisted of 4 with 98 students, 50 students (26 and 24) in the spring 2017 semester and 48 students (23 and 25) in the fall 2017 semester, engaging with equivalent numbers of persons with disabilities from local non-profit organizations in developing personalized Web sites. Few of the students engaged individually with persons with disabilities, and few of the persons with disabilities ever engaged one-on-one with students without disabilities, until all of them joined the projects of the semesters, which was learned at the beginning of the semesters from non-profit organizational and school surveys. Most of the students were experienced in skills on the Web, a foundation for interfacing with the persons with disabilities, which was a goal more important than mere Web design (McNeil, 2012). They were not experienced with the Mahara 17.04 e-Portfolio system in the Seidenberg School, in its customized features of blogs for documenting engagement progress after each class, a context of critiques for documenting impressions of societal stories after each class and essay journals for documenting partnership reflections at mid-semester and at end of the semester – requirements of pure writing - and displays of project Web sites (Hand, Kent & Bell, 2012; Adams, Blumenfeld, Castaneda, Hackman, Peters & Zuniga, 2000). Most of the undergraduate students were fulfilling the projects merely as a requirement of outreach services in the university. The courses were 3 hours 1 day a week for 14 semester weeks, with gala presentations of the Web sites in the school on the 14th week.

The benefits of documentation in the e-Portfolios of the experiences with the persons with disabilities may not be evident however in facilitating learning outcomes. The features of the e-Portfolios may not be facilitating engagement and advocacy of the students with the persons with disabilities, as learning outcomes of the projects. The meaningfulness of the projects in the semesters, beyond showcases of Web sites, may not be in the increased recognition of the responsibility of service to those with developmental and intellectual disabilities (Prentice & Garcia, 2000). The recognition of service, through the setting of technology in schools of information systems for those with disabilities, is nevertheless noted in the literature (Hoxmeier & Lenk, 2003). The paper attempts to evaluate the facilitating impacts or non-impacts of the e-Portfolios on increased or non-increased engagement and advocacy outcomes of the services of the students for those with disabilities.

3. FOCUS OF PAPER

The paper evaluates the e-Portfolios as to facilitating features of the Mahara 17.04 e-

Portfolio system, in impacting or not impacting the perceptions of the students as to the people with disabilities. The functionality of the system may be an important ingredient in the learning outcomes of the students (i.e. in learning the potential of the people with disabilities and in learning the potential of technology to help the people with disabilities to express themselves), as they record reflections on their service to the people with disabilities (Morreale, Van Zile-Tamsen, Emerson & Herzog, 2017). To evaluate the reflections of the students, the paper is focused on the following:

Engagement
Importance – e-Portfolio features facilitating or not facilitating learning of the students of the potential of those with disabilities, on the projects in Web Design for Non-Profit Organizations; and
Satisfaction – e-Portfolio features facilitating or not facilitating performance satisfaction of the students of those with disabilities, on the projects.

Advocacy
Self-Efficacy – e-Portfolio features facilitating or not facilitating a foundation of learning of the students to be advocates of those with disabilities in society; and
Sociality – e-Portfolio features facilitating or not facilitating a motivation of the students in Web Design for Non-Profit Organizations to be involved in passionate proactive service with those with disabilities in society.

These measurements of reflections in the e-Portfolio system were methods in previous progressing studies by the authors of students in the Seidenberg School of Computer Science and Information Systems (Lawler, 2013; Lawler, Iturralde, Goldstein & Joseph, 2015).
The paper is attempting to learn from the documentation sections of the students if the e-Portfolio system is facilitating the civic learning outcomes of the projects of the students. There are few papers focused on e-Portfolios facilitating or not facilitating justice for the rights of those with disabilities to be helped by the technology of the Web (Braddock, Hoehl, Tanis, Ablowitz & Haffer, 2013, p.95; Richards-Schuster, Ruffolo, Nicoll, Distelrath & Galura, 2014), which will be a contribution of this paper.

4. METHODOLOGY OF PAPER

The methodology of the paper evaluated the courses of Web Design for Non-Profit Organizations (4) in the Seidenberg School of Computer Science and Information Systems of Pace University.

The experiences of the students (n=98 [n=26 and 24 in spring 2017 and n=23 and 25]) in fall 2017) on the partnered projects (i.e. www.wix.com) with the persons with disabilities were evaluated from the blogs (14 entries per student), critiques (14 entries per student) and journals (2 entries of multiple pages per student) recorded in the e-Portfolio system and were summarized by the first author, at the end of the semesters. The experiences of the students (n=50 and 48) with generic persons with disabilities were also evaluated in an instrument of survey from the beginning of the spring and fall 2017 semesters for comparison, at the end of the semesters. The documentation of the experiences of the students (n=98), from interactions on the projects with those moderately impaired with disabilities as recorded in the e-Portfolio system, was individually interpreted in engagement – *importance* and *satisfaction* and in advocacy – *self-efficacy* and *satisfaction* by the first author, with aide assistance, on a high (5) to low (1) impact or zero scaling, from content measurement principles and standards of content validity (Neuendorf, 2017) of key phrasing and key wording and was independently re-interpreted collectively by the second author of the paper, at the end of the semesters. The experiences of a focus group (Krueger & Casey, 2009) of a random sample of the students (n=18 [n=8 in spring and 10 in fall 2017]) was interpreted by both authors, as a final measurement of the paper.

The methodology of the paper was similar to the previous studies by the authors of students in the Seidenberg School (Lawler, 2013 & Lawler, Iturralde, Goldstein & Joseph, 2015).

The interpretation of statistics (Frankfort-Nachmias & Leon-Guerrero, 2015) was performed by the second author, from Microsoft EXCEL 2010 and IBM SPSS Statistics 24, for presentation in the next section of the study.

5. ANALYSIS OF DATA AND DISCUSSION OF RESULTS

The e-Portfolios in Web Design for Non-Profit Organizations are found to be facilitating the engagement (means=4.33/5.00) and the advocacy (2.98/5.00) of the students for people with cognitive disabilities. Favorable features of reflections in the customized but regimented sections of the system (Swan & Hicks, 2007) are helping in engagement – *importance* (4.19/5.00) and *satisfaction* (4.48 /5.00) and in advocacy – *self-efficacy* (3.21/5.00) and *sociality* (2.76/5.00) of the students, from their information systems projects with the people with disabilities, as indicated in Table 1 of the Appendix. The findings are especially important, as most of the students (96 /98 or 97%) had not met moderately impaired but nimble people with disabilities previously, and most of them (77/98 or 78%) had not had mechanisms in the school for recording personal perspectives or reflections on projects of technology, and notably not for this new population.

e-Portfolio Blogs – Reflections on Engagement or Project Progress

The blogs of the students, in which they are recording experiences in reflections on their engagement or project progress after each class in the semesters, are indicating engagement (4.19/5.00) - in *importance* (3.92/5.00) and *satisfaction* (4.47/5.00) and advocacy (2.89/ 5.00) - in *self-efficacy* (3.01 /5.00) and *sociality* (2.77 /5.00) for the people with disabilities, in Table 2. The students are recording generally favorable relationships with their partnered persons with disabilities. The progression of the projects and the relationships is noted in the reflections of the students.

e-Portfolio Critiques – Reflections on Generic Societal Stories

The critique entries of the students, in the school of computer science and information systems, in which they are recording experiences in reflections on generic societal stories of those with disabilities after each class, are indicating engagement (4.28/5.00) – in *importance* (4.20 /5.00) and *satisfaction* (4.36 /5.00) and advocacy (2.88/5.00) - in *self-efficacy* (3.05/5.00) and *sociality* (2.70/5.00) for those with disabilities, in Table 3.

e-Portfolio Journals – Reflections on Partnered Relationships and Project Results

The essay journal pages of the students, in which they are recording experiences in reflections on their partnered relationships and project results, at mid-semesters and at the end of the semesters, are indicating engagement (4.53/5.00) – *importance* (4.45/5.00) and *satisfaction* (4.60/5.00) and advocacy (3.18/5.00) - in *self-efficacy* (3.56/5.00) and *sociality* (2.80/5.00) for those with disabilities, in Table 4.

Beyond the blog entries and the critique entries, the essay journals with multiple pages were of the highest insight from the reflections at mid-semesters and at the end of the semesters. The journals required of the students more reflective thinking and were of most value.

e-Portfolio Blogs, Critiques and Journals

Most of the students are collectively reporting increased *importance* (4.19/5.00) in the "brain diversity" potential (Wille & Sajous-Brady, 2018) of those moderately impaired but nimble persons with disabilities, to be further proficient with information systems tools, and they are collectively reporting increased pride in *satisfaction* (4.48/5.00) in the semesters with them. They are "pausing to reflect" (Miller & Morgaine, 2009, p.6) in the e-Portfolio mechanism on their personal results (Strang, 2015) of servicing those with disabilities with the design methodologies and the technologies of the Web, and they are sharing their "success of their teaching" (Buyarski, Aaron, Hansen, Hollingsworth, Johnson, Kahn & Powell, 2015) with students on the other teams. At the same time, the entrepreneurial persons with developmental and intellectual disabilities are indicating pride in satisfaction and in success with their new Web sites, their partnered students and themselves, which may prompt them to pursue further support with the technology of the Web at the university (Ferrette, 2018).

Summary

The findings from the e-Portfolio blogs, critiques and journals are collectively indicating engagement (4.33/5.00) as higher than advocacy (2.98 /5.00) as the learning outcomes of the projects of the students, facilitated by the implementation of the e-Portfolio instructional pedagogy in the sections of the system (Tang & Austin, 2009). The engagement finding may be from excitement of the immediate project results of Web sites; and the advocacy finding may be from impressions of project relationships to be hopefully later nourished from the overall results of the semesters (Flavell, 1979). The foundation however for public service to marginalized populations of society, such as those with disabilities, is latent in the reflections of the students.

The findings from the semesters of spring 2017 to fall 2017 are indicating nevertheless engagement and advocacy progression of the students with those with disabilities.

Other findings from a focus group of a random sample (n=18) of the students are indicating engagement (4.64/5.00) – *importance*

(4.57/5.00) and *satisfaction* (4.70 /5.00 and advocacy (3.53/5.00) – *self-efficacy* (3.70/5.00) and *sociality* (3.35/5.00) results similarly of the other (n=80/98) students, detailed from the sample summary in Table 5.

A random reflection sampling of statements of the students in Web Design for Non-Profit Organizations is in Table 6.

In summary, the e-Portfolios are an evidently facilitating system, for the favorable recording and sharing of the reflections and the solutions of the neuro-typical students on information systems projects with neglected neuro-atypical people with disabilities, with whom they would normally not be in any relationships in semesters in a university.

(e-Portfolio statistics, including correlations and frequencies in Tables 7-8, are in the Appendix of the study.)

6. IMPLICATIONS OF STUDY

The features of the e-Portfolios are facilitating the engagement and advocacy experiences of the students. The evidence of learning outcomes are in the reflections of the information systems students, as reported on the e-Portfolios. From blogs to journals, the e-Portfolios are easily guiding the learning of the students as they are recording reflections on the people with disabilities. The e-Portfolios are essentially the learning spaces identified with the students (Grush, 2016). The implication for instructors in information systems is that e-Portfolios may be helpful in improving learning outcomes of students on projects of service.

The features of the e-Portfolios are facilitating interactions with other students. The blogs, critiques and journals, as open to other students on the system, are illuminating interrelationships with frequently perplexing but human people with disabilities to millennial students initially involved with this population (Landis, Scott & Kahn, 2015). The implication for instructors is that e-Portfolios may be helpful in increasing learning outcomes from the social spaces of the students.

The e-Portfolios are helping in the learning of people different than the students: people with cognitive disabilities. The projects are involving the students with moderately impaired people with disabilities, a population they had not met previously, on projects of technology. These projects are importantly involving the students in progressive reflections (Reynard, 2009) on the service to those with disabilities –reflections on "something … students [will not] forget" (Mummalaneni, 2014), as they might forget on other projects of technology. The e-Portfolios are the mechanisms for presentations and for reflections on the project results of the students (Holland, 2015). This implication may be helpful for instructors in information systems in initiating learning outcomes represented in e-Portfolios not only by project results but also by the reflections of the students.

The e-Portfolios may be further helpful on other outreach projects of service. Information systems students impacted positively by the projects with the people with developmental and intellectual disabilities may be motivated to be on other projects of the Web with other neglected populations of society, or even with students with disabilities already in the university, on which they may be recording reflections in e-Portfolios or in other forums of service, such as wikis (National Council of Teachers of English, 2015; Plownan, 2007). This implication may be helpful to instructors integrating e-Portfolios more on other projects of public service involving technologies.

Lastly, the benefits of the e-Portfolios were not only for the students but also for the persons with disabilities partnered with them. The non-profit organizational staff indicated that the persons with disabilities learned the potential of socialization with those without disabilities, an implication in lessening frequent isolation of this group (Boucher, 2017; Mazurek, 2014). They learned the potential of the Web with the results of the Web sites for themselves. Though the students learned moreover the potential of the Web for those with disabilities, they learned importantly the meaningfulness of the rights of those with disabilities, as noted in the e-Portfolio reflections, to the technologies of the Web, which if spread by the students helps others of those with disabilities (Hoy, 2018; Braddock, Hoehl, Tanis, Ablowitz & Haffer, 2013, p. 98). This implication is important to instructors in information systems in integrating pedagogy involving reflections of students, and not limiting the pedagogy to mere project results, through systems such as e-Portfolios.

7. IMITATIONS AND OPPORTUNITIES IN RESEARCH

The paper is a descriptive study of students at one university. The perceptions of the students in their reflections may be inflated by mere pleasure in the project results of the technology. The persons with disabilities in mentorships or partnerships with the students were with developmental and intellectual disabilities, not other disabilities, and they were moderately impaired in disorders not high or low on the spectrum. These limitations may have impacted project relationships and technologies. The results may not be generalized immediately without caution.

The paper may nevertheless be helpful to instructors in information systems in integrating e-Portfolio interdisciplinary projects of service. The potential of nimble persons with disabilities to be on projects, such as Web Design for Non-Profit Organizations, justify a productive role for them (Fitch, Peet, Glover & Tolman, 2008). The e-Portfolio reflections are promising for future public service of information systems students with those with disabilities. In a future study, there may be less laborious but more mining of the reflections of the students with sentiment software tools (Liu, 2012). In short, the results of this paper recommend further study with the e-Portfolio system.

8. CONCLUSION

This paper encourages e-Portfolios on course projects of service in schools of computer science and information systems.

e-Portfolios are facilitating the learning of net generation students of the potential of people with cognitive disabilities to be proficient with technology. Favorable impacts are found in the e-Portfolios in increased engagement and increased advocacy for the people with disabilities, from the narrative reflections of the students. Features of the e-Portfolios are essentially found to be especially helpful in learning outcomes on the projects of service of the neuro-typical students. Functionality furnished mechanisms for more recording of reflections on results of service and more sharing of success than other systems. Students in Web Design for Non-Profit Organizations were often from a personal perspective recording reflections and solutions in their e-Portfolios on moderately impaired neuro-atypical people with disabilities, with whom they were partnering in the semesters, the writing of which is not often the results on other course projects in schools of information systems. This was important in the learning outcomes of the students, in the remembrance in their e-Portfolios of their services to people with disabilities through the technology of the Web.

Therefore, this paper will be inspirational to instructors involving students on reflections and solutions of technologies with neglected but nimble populations of society, and their integration of e-Portfolios will promote the public service of schools of computer science and information systems.

9. REFERENCES

Accardo, P.J., & Whitman, B.Y. (2011). Developmental Disabilities Terminology. Paul H. Brookes Publishing Company, Baltimore, Maryland.

Adams, M., Blumenfeld, W.J., Castaneda, R., Hackman, H.W., Peters, M.L., & Zuniga, X. (2000). Readings for Diversity and Social Justice. Routledge, Oxford, United Kingdom.

Amaya, P., Agudo, J.E., Sanchez, H., Rico, M., & Hernandez-Linares, R. (2013). Educational e-Portfolios: Uses and tools. *Social and Behavioral Sciences*, 93, 1169-1173.

Boucher, J. (2017). Autism Spectrum Disorder: Characteristics, Causes and Practical Issues. Sage Publications, Inc., Thousand Oaks, California.

Braddock, D., Hoehl, J., Tanis, S., Ablowitz, E., & Haffer, L. (2013). The rights of people with cognitive disabilities to technology and information access. *Inclusion*, 1(2), 95,98.

Buyarski, C.A., Aaron, R.W., Hansen, M.J., Hollingsworth, C.D., Johnson, C.A., Kahn, S. & Powell, A.A. (2015). Purpose and pedagogy: A conceptual model for e-Portfolio. *Theory into Practice*, 54(4), 283-291.

Clark, E., & Eynon (2009). e-Portfolios at 2.0: Surveying the field. *Association of American Colleges and Universities*. 11(1), Winter, 1-2.

Dahstrom, E. (2015).Davis, V. (2015, April 30). Eleven Essentials for excellent e-Portfolios. *Edutopia*, 2.

Exter, K.D., Rowe, S., Boyd, W., & Lloyd, D. (2012).Using Web 2.0 technologies for collaborative learning in distance education: Case studies from an Australian university. Future Internet, 4, 217.

Eynon, B., & Gambino, L.M. (2017). High-Impact e-Portfolio Practice: A Catalyst for Student, Faculty, and Institutional Learning. Stylus, Sterling, Virginia, 42,69.

Ferrette, C. (2018, March 23). More colleges create support, opportunity for students with autism. *Disability Scoop*, 1-4.

Fitch, D., Peet, M., Glover, B., & Tolman, R. (2008). The use of e-Portfolios in evaluating the curriculum and student learning. *Journal of Social Work*, 44(3), 37-54.

Flavell, J.H. (1979). Metacognition and cognitive monitoring: A new area of cognitive developmental inquiry. *American Psychologist*, 34, 906-911.

Frankfort-Nachmias, C., & Leon-Guerrero, A. (2015).Social Statistics for a Diverse Society. Sage Publications, Inc., Thousand Oaks, California.

Garrison, W., & Ring, G. (2013). Walking the e-Portfolio talk: Begin your e-Portfolio in focus hours. *Association for Authentic Experiential and Evidence-Based Learning*, Workshop, Boston, Massachusetts.

Grush, M. (2016). E-Portfolio: The learning management system (LMS) for students? *Campus Technology*, February 16, 3.

Hand, R., Kent, D., & Bell, T.W. (2012). Mahara e-Portfolios: Beginners Guide. Packt Publishing, Ltd., Birmingham, United Kingdom.

Holland, B. (2015, June 30). The art of reflection. *Edutopia*, 1,3.

Horowitz, S., Rawe, J., & Whittaker, M. (2017). The State of Learning Disabilities: Understanding the One in Five. National Center for Learning Disabilities, New York, New York.

Hoxmeier, J., & Lenk, M.M. (2003). Service-learning in information systems courses: Community projects that make a difference. *Journal of Information Systems Education (JISE)*, 14(1), 91.

Hoy, A.Q. (2018, February 5). Integrating human rights with STEM education inspires students. *American Association for the Advancement of Science* (AAAS), 1-6.

Krueger, R.A., & Casey, M.A. (2009). Focus Groups: A Practical Guide for Applied Research. Sage Publications, Inc., 4th Edition, Thousand Oaks, California, 85-108.

Laird, T.F., Shoup, R., & Kuh, G.D. (2005). Measuring deep approaches to learning using the National Survey of Student Engagement. *Annual Meeting of the Association for Institutional Research*, Chicago, Illinois.

Landis, C.M., Scott, S.B., & Kahn, S. (2015). Examining the role of reflection in e-Portfolios: A case study. *International Journal of e-Portfolio*, 5(2), 107.

Lawler, J.P. (2013). Engaging community service students through digital portfolios. *Information Systems Education Journal (ISEDJ)*, 11(1), February, 41-54.

Lawler, J., Iturralde, V., Goldstein, A. & Joseph, A.(2015).Engaging engineering and information systems students in advocacy for individuals with disabilities through a disability film media project. *Information Systems Education Journal (ISEDJ)*, 13(4), July, 51-62.

Light, T.P., Chen, H.L., & Ittelson, J.C. (2012). Documenting Learning with e-Portfolios: A Guide for College Instructors. Jossey-Bass, San Francisco, California, 46, 87-88.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1).

Lorenzo, G., & Ittleson, J.C. (2005) Demonstrating and Assessing Student Learning with e-Portfolios. Educause Learning initiative, Boulder, Colorado.

Mazurek, M.O. (2014). Loneliness, friendship, and well-being in adults with autism spectrum disorders. *Autism*, 18(3), 223-232.

McNeil, P. (2012). The Designer's Web Handbook: What You Need to Know to Create for the Web. HOW Books, Cincinnati, Ohio.

Miller, R., & Morgaine, W. (2009). The benefits of e-Portfolios for students and faculty in their own words. *Association of American Colleges and Universities*, 11(1), Winter, 2,6.

Milsom, A. (2017, October 10). Creating positive school experiences for students with disabilities. *Reading Rockets*, 1.

Morreale, C., Van Zile-Tamsen, C., Emerson, C.A., & Herzog, M. (2017). Thinking skills by design: Using a capstone e-Portfolio to promote reflection, critical thinking, and curriculum integration. *International Journal of e-Portfolio*, 7(1), 20.

Mummalaneni, V. (2014). Reflective essay and e-Portfolio to promote and assess student learning in a capstone marketing course. *Marketing Education Review*, 24(1), 43-46.

Neuendorf, K.A. (2017). The Content Analysis Guidebook. Sage Publications, Inc., 2nd Edition, Thousand Oaks, California, 124-128,297-298.

Plowman, T.S. (2007). Wikis as a social justice environment. *Proceedings of the Society for Information Technology and Teacher Education International Conference*, Chesapeake, Virginia, 749-751.

Prentice, M., & Garcia, R.M. (2000). Service-learning: The next generation in education. *Community College Journal of Research and Practice*, 24(1), 19-26.

Reynard, R. (2009). More challenges with wikis: Four ways to move students from passive to active. *The Journal.Com*, October, 1.

Reynolds, C., & Patton, J. (2014). Leveraging the e-Portfolio for Integrative Learning: A Faculty Guide to Classroom Practices for Transforming Student Learning. Stylus, Sterling, Virginia, 42.

Richards-Schuster, K., Ruffolo, M.C., Nicoll, K.L., Distelrath, C., & Galura, J.A. (2014). Using e-Portfolios to access program goals, integrative learning, and civic engagement: A case example. *International Journal of e-Portfolio*, 4(2), 133.

Salisbury, C.L., Gallucci, C., Palombardo, M.M, & Peck (1995). Strategies that promote social relationships among elementary students with and without severe disabilities in inclusive schools. *Exceptional Children*, 62, 125-137.

Strang, T. (2015, April 29). Three reasons e-Portfolios matter to today's college students. *Cengage*, 2.

Swan, K., & Hicks, D. (2007). Through the democratic lens: The role of purpose in leveraging technology to support historical thinking in the social studies classroom. *The International Journal of Social Studies Education*, 21(2), 142-168.

Tang, T. L-P., & Austin, M.J. (2009). Students' perceptions of teaching technologies, application of technologies, and academic performance. *Computers and Education*, 10(1016).

Wehmeyer, M.L. (2013). The Story of intellectual Disability: An Evolution of Meaning, Understanding, and Public Perception. Paul H. Brookes Publishing Company, Baltimore, Maryland.

Wille, S., & Sajous-Brady, D. (2018). The Inclusive and accessible workplace. *Communications of the ACM*, 61(2), February, 24.

Yancey, K.B. (2009). Electronic portfolios a decade into the twenty-first century: What we know, what we need to know. *Association of American Colleges and Universities*, Peer Review, 11(1), Winter, 2.

_____ (2015). Principles and practices in electronic portfolios: Position statement. National Council of Teachers of English (NCTE), March, 1.

APPENDICES

Table 1: e-Portfolio Engagement and Advocacy Perceptions* – Consolidated Findings from Blogs, Critiques and Journals of Information Systems Students - Spring 2017 – Fall 2017

	MEAN	STANDARD DEVIATION
ENGAGEMENT	4.33	1.03
Importance	4.19	1.12
Satisfaction	4.48	0.90
ADVOCACY	2.98	1.79
Self-Efficacy	3.21	1.70
Sociality	2.76	1.85

*n=98 students (Tables 1- 4 and 6-8)

Table 2: e-Portfolio Engagement and Advocacy Perceptions – Findings from Blogs (on Engagement or Project Progress with Persons with Disabilities) of Information Systems Students – Spring 2017 – Fall 2017

	MEAN	STANDARD DEVIATION
ENGAGEMENT	4.19	1.22
Importance	3.92	1.37
Satisfaction	4.47	0.98
ADVOCACY	2.89	1.83
Self-Efficacy	3.01	1.80
Sociality	2.77	1.85

Table 3: e-Portfolio Engagement and Advocacy Perceptions – Findings from Critiques (on Generic Societal Stories of Persons with Disabilities) of Information Systems Students – Spring 2017 – Fall 2017

	MEAN	STANDARD DEVIATION
ENGAGEMENT	4.28	1.00
Importance	4.20	1.04
Satisfaction	4.36	0.96
ADVOCACY	2.88	1.85
Self-Efficacy	3.05	1.82
Sociality	2.70	1.87

Table 4: e-Portfolio Engagement and Advocacy Perceptions – Findings from Essay Journals (on Partnered Relationships and Project Results with Persons with Disabilities) of Information Systems Students – Spring 2017 – Fall 2017

	MEAN	**STANDARD DEVIATION**
ENGAGEMENT	4.53	0.80
Importance	4.45	0.84
Satisfaction	4.60	0.76
ADVOCACY	3.18	1.67
Self-Efficacy	3.56	1.41
Sociality	2.80	1.83

Table 5: e-Portfolio Engagement and Advocacy Perceptions – Findings from Focus Group of Information Systems Students - Spring 2017 – Fall 2017**

	BLOGS		**CRITIQUES**		**JOURNALS**	
	MEAN	**STANDARD DEVIATION**	**MEAN**	**STANDARD DEVIATION**	**MEAN**	**STANDARD DEVIATION**
ENGAGEMENT	4.25	0.97	4.78	0.59	4.89	0.40
Importance	4.11	1.02	4.72	0.67	4.89	0.32
Satisfaction	4.39	0.92	4.83	0.51	4.89	0.47
ADVOCACY	3.81	1.35	3.42	1.79	3.36	1.84
Self – Efficacy	4.06	1.00	3.61	1.61	3.44	1.82
Sociality	3.56	1.62	3.22	1.99	3.28	1.90

**n=18 students

Table 6: Random Sampling of Semester Statements of Information Systems Students - Spring 2017 – Fall 2017

Spring 2017

Before the course, I did not consider people with disabilities at all … course was a great experience for me.

… Did a Web site closely with a person with disabilities … learned people with disabilities are humans … learned of the discrimination … learned how to fight for them … proud to say this was one of the most informative and interesting courses I have ever taken [in the university].

Eye-opening experience … never guessed I would be helping a person with disabilities … improved my interpersonal skills … inspired me to look into what it is like living with intellectual disabilities … reflecting and writing were new to me.

… learned not to be closed-minded … but open to people with disabilities … open to situations … students learned a lot with them … I am a better person from the project with them.

… learned more about life and about people than from any of the courses in the school … more observant [of people with disabilities] … more passionate about them … will stay with me forever … who would have thought [Web Design for Non-Profit Organizations] would impact me so positively?

Fall 2017

… learned that there are no differences … between people with or without disabilities … people with disabilities should be introduced to technologies … experience was great … not easy or hard but rewarding … something you do not often see [in the school].

… never interacted with a person with intellectual disabilities … for such an extended period … greatest takeaway was how to interact with others … knowledge of the disability movement … through the readings.

… learning outcome not what I expected in the school … learned my own problems are insignificant … I want to be involved more on projects like this.

I cannot describe it – You have to experience it … You cannot get this feeling in any other courses [in the university] in my opinion … would not have imagined not taking this … would not have imagined writing about this every week.

… a unique experience I will not forget … helped me grow as a person.

Table 7: Non-Paramertic Kendall's tau-b Correlations of Study

	ENGAGEMENT		**ADVOCACY**	
MEASUREMENTS (VARIABLES)	IMPORTANCE RATINGS	SATISFACTION RATINGS	SELF- EFFICACY RATINGS	
SATISFACTION RATINGS	0.603*			
SELF- EFFICACY RATINGS	0.478*	0.478*		
SOCIALITY RATINGS	0.383*	0.370*	0.723*	

*Correlation is significant at the 0.01 level (2-tailed).

Table 8: Frequency Distributions of Study

	ENGAGEMENT		**ADVOCACY**	
MEASUREMENTS	IMPORTANCE	SATISFACTION	SELF-EFFICACY	SOCIALITY
5 - Very High Impact	169 (57%)	206 (70%)	97 (32%)	80 (27%)
4 - High Impact	42 (14%)	28 (9%)	25 (8%)	7 (2%)
3 - Intermediate Impact	68 (23%)	54 (18%)	112 (38%)	122 (41%)
2 - Low Impact	4 (2%)	2 (1%)	8 (4%)	2 (2%)
1 – Very Low Impact	7 (2%)	2 (1%)	6 (3%)	12 (4%)
0 – No Impact or Blank	4 (2%)	2 (1%)	46 (15%)	71 (24%)
	294	294	294	294

Note: 294=98 (Students) x 3 (Blogs, Critiques and Journals) for Distribution Purposes

A Cross Collegiate Analysis of the Curricula of Business Analytics Minor Programs

Timothy Burns
tburns1@ramapo.edu

Cherie Sherman
csherman@ramapo.edu

Anisfield School of Business
Ramapo College of New Jersey
Mahwah, NJ 07430

Abstract

In recent years, there has been an explosion in the demand for personnel with analytics skills. Given that the demand for this skill set cuts across so many disciplines, it is a useful addition to any major and an ideal candidate as an academic minor. Furthermore, as the underlying analytics tools and techniques used by data analysts emerged primarily from the business disciplines, the school of business makes an ideal place to house an analytics minor. Given the high demand for business analytics skills and the ubiquitous nature of the analytics field, the question becomes what topics should be included in the curriculum of a business analytics minor? The goal of this research is to answer that question by analyzing the curricula currently offered by a large sample of business analytics minor programs.

Keywords: Business Analytics Knowledge and Skills, Business Analytics Minor Curriculum

1. INTRODUCTION

In recent years, there has been an explosion in the demand for personnel in the field of analytics. Particularly for professionals with the knowledge, skills and competencies required for the methodical exploration and investigation of data. Using descriptive, predictive, and prescriptive statistical tools, analytics professionals support decision making and gain insight into business performance. Ranked second in a Computerworld survey on the most difficult skills to find, analytics expertise is scarce (Computerworld 2018). McKinsey Global Institute reports that the United States could face a shortage of between 140,000 and 190,000 individuals who possess business analytics skills and an additional 1.5 million managers with the skills to implement the results (McKinsey Global Institute 2018). In addition, according to the Bureau of Labor Statistics, employment in the data analytics area is projected to grow by 27 percent by 2022 (U.S. Bureau of Labor Statistics 2018).

An understanding of analytics has the potential to add value to almost any career path as analytics spans across all disciplines and industry sectors. Students with these skills are in high demand in a variety of industries and sectors including accounting, management, marketing, finance, information systems, operations, health care, human resources, engineering, politics, sports, energy, etc. Given that the demand for this skill set cuts across so many disciplines, it is a useful addition to any major and an ideal candidate as an academic minor. Furthermore, as the underlying analytics tools and techniques emerge from disciplines such as management science, operations research, statistics, business intelligence, information systems, and traditional business fields, the school of business makes an ideal place to house an analytics minor.

Given the high demand for business analytics skills and the ubiquitous nature of the analytics field, the question becomes what topics should be included in the curriculum of a business analytics minor? The goal of this research is to answer that question by reviewing, analyzing, and tabulating the curricula currently offered by a large sample of business analytics minor programs. Although Phelps and Szabat's research (Phelps and Szabat, 2015) indicates 73% of academics polled believe a major or minor degree option will be offered by most major business schools in the near future, the minor has gotten little attention.

2. LITERATURE REVIEW

The literature related to data analytics education is still scant and in particular, very little is devoted to the subject of this paper, the data analytics minor. Some of the difficulty in identifying the literature is the lack of uniformity in the use of the terms "business intelligence," "business analytics," and "decision science." Zheng (2018) defined analytics as a catch-all term which is an expansion of business intelligence and which currently refers to the generalized leveraging of information for more strategic decision-making. Likewise, he sees data science as an interdisciplinary field which has evolved from business intelligence, using models and methods to create insights from data. Rostcheck (2016) draws a finer distinction between business intelligence and data science characterizing business intelligence as backwards-looking, analytic, and data base focused and data science as predictive due to its incorporation of hypotheses and modeling. Business/Data Analytics programs are most often housed in business departments and data science programs are typically offered through computer science departments or are interdisciplinary (Philips and Szabot, 2015, p. 2).

Wang (2015) identified 44 research papers that were pertinent by searching Google Scholar for titles of articles published in journals or conference proceedings between 2009 and 2015 using keywords such as "business intelligence" and "analytics education." After retrieving 26 papers, he then used snowballing and citation analysis, suggested by Boell and Cecez-Kecmanovic, to find further relevant literature. As a result, he hand-selected an additional 18 papers that met the criteria. He then categorized the papers into the five research categories they covered. These included: The Current State of Analytics Education; The Role of the Data-savvy Professional; The Design of the Analytics Curriculum; Teaching Cases and Learning Activities; and Skill Sets and Knowledge Domains of Professionals.

Wang identified five areas that were not addressed in the research papers. Firstly, although the perspectives of faculty, students, and practitioners were addressed, feedback from new graduates had not been. Also, learning activities in the curricula seemed to be exploratory or case study in nature but theoretical frameworks or empirical methods for assessing student learning were not discussed. The literature discussed only analytics education in the United States rather than global education. Pedagogical innovations and the design of teaching courses was not given much attention. Finally, there was no articulated framework of knowledge domains and skills sets for the analytics professional.

Phelps and Szabat (2015) surveyed approximately 100 academics with a connection to statistics education to better understand how undergraduate institutions were preparing to develop data analytics programs. The survey questions were developed based upon earlier research by Aasheim, Runter, and Gardiner (2014) and Gorman and Klimberg (2014). These studies found that analytics programs were not just a rebranding of existing curricula but were novel. The programs were heterogeneous and dependent upon expertise of faculty, type of student, and local industry. The authors noted that there was an interest in integrating analytics throughout the curriculum.

Phelps and Szabat (2015) found that 53% of schools did not yet offer an undergraduate major in business analytics or decision sciences and 59% did not offer a minor. Fifty-nine percent offered neither a business analytics or decision science minor but 6% offered both, 6% only at the graduate level and 9% only at the undergraduate level. About 30% of the respondents were considering creating a business analytics or decision science major and 26% were considering a minor. Most programs required only one statistics course and Excel was the software of choice for statistics, R for business analytics. Statistics and Information Systems technology faculty had the greatest involvement in teaching the courses.

Wilder and Ozgur (2015) identified three levels of skills which the data analytics undergraduate curriculum should address. The data scientist should have an advanced degree in a quantitative discipline and a foundation in computer science and mathematics which includes probability and statistics. Data scientists work on more challenging problems and can find employment as consultants or as part of an internal corporate specialized group. At the next level, the data specialist should understand how data is stored and how to access and analyze it. Data specialists may work in a centralized IT support group or be scattered throughout the functional business areas. At the final level, the business analyst is a data-saavy manager who understands how to identify and frame the business problem or question to be addressed/answered through data analysis. Citing Watson (2013), Wang also maintained that three types of analytics should be addressed: descriptive (what has already happened); predictive (what is expected to happen) and prescriptive (what should happen).

Wilder and Ozgur (2015) suggested that graduate curricula should be used as a model for undergraduate studies and noted that in 2015, there were 49 such programs. They also noted that the Institute for Operations Research and the Management Science (INFORMS) offers a Certified Analytics Professional (CAP) exam which provides a guide for curriculum development. They mentioned that minors and certifications should be considered as other options.

The curriculum they proposed for the business analyst has six major courses and one core course and is organized around five knowledge domains: project life cycle, data management, analytical techniques, deployment, and a functional area. These domains are to be spread across the curriculum rather than provided in individual, specialized courses. For example, traditional business math courses such as calculus and statistics would need to be updated to incorporate these knowledge areas. The six required courses include Data Management (tools such as SQL), Descriptive Analysis (statistics), Data Visualization (key indicators, scorecards, dashboards), Predictive Analytics (advanced statistics), Prescriptive Analytics (Spreadsheet Models), and Data Mining (CRISP-DM). A Practicum and Electives complete the major. They recommended an overall approach which leads the student from the specific (real life problem) to the general (how to use decision trees), to maintain student interest and ensure that students view the coursework as relevant.

Wymbs (2016) referred to the Burning Glass Technologies Report for identifying the skill sets for Data Analytics jobs which included statistical package skills (SPSS, SAS, R) Microsoft PowerPoint and Excel, financial skills such as risk management, some programming skills, and critical thinking and communications skills. He noted that as of 2016, there were 517 data science/data analytics programs of which 374 were Master’s programs, 88 Certificate programs, 36 Bachelor’s programs, and 10 PhD programs. A query of the AACSB database of 1,500 institutions indicated only 11 undergraduate programs in data analytics and data science but 56 programs in business analytics. Some of these may have also have been implemented by changing just the names of existing courses and programs rather than by truly innovating. Wymbs cited Wang (2015) and Wilder and Ozgur (2015) for the proposition that most business analytics programs are in the Business School, are in their infancy, lack a commonly accepted curriculum model and do not include a design for developing students’ professional skills. Unanswered questions include: how many courses should be in a BI/BA major or minor?

Wymbs (2016) also discussed insights provided by the 2015 Business Higher Education Forum Conference attended by investment banks, accounting firms, publishers, tech companies, and the Federal Reserve. Participants preferred “R” and Python as programming languages and indicated that they expected new graduates, particularly accountants, to have data analytics proficiency in ordered to be hired. This led to the author’s institution offering a business minor in data analytics and tracks in CIS and Marketing which could be completed within one year. Important ongoing contributions from the business community included real datasets for data mining, internships, case studies, and advice. For schools who want to initiate analytics education earlier, Temple University offers a General Education data analytics course, without prerequisites, which can be used as a model.

Meyer (2015) pointed out that even INFORMS states that there is no consensus as to what constitutes a defined curriculum for data analytics. Meyer sees the subject as multi-disciplinary and described a cross-college undergraduate program, developed at Drake College, wherein the student could earn a data analytics degree in either the College of Arts and Sciences or the College of Business. The core of the program consisted of computer science and statistics along with a specialty area such as marketing or bioinformatics. Meyer concluded that practically speaking, the elements of data analytics are: data/database, statistics, operations research, computer science, and managerial strategy. As these courses already exist, schools who develop a data analytics program add courses such as Data Visualization, Programming in R, or Customer Sentiment Analysis to establish an immediate presence in the field.

Meyer described the contents of a sample of data analytics programs in the Midwest which included undergraduate, graduate, and certificate programs. Foci ranged from Management, to Information Systems, to Operations Research. In addition, he described a minor at the University of Cincinnati which included two courses in Business Analytics, a course in Data Mining and a course in Spreadsheet Analysis. Elective Courses were far ranging as to topic and included: Forecasting, Data Visualization, Econometrics, Database Design, Financial Modeling, and Marketing Research. He also described a Data Analytics minor at Drake which included two courses in calculus, two courses in computer programming, an introduction to R and SAS, Statistics, Cloud Computing or Database, Machine Learning, and Data Mining or Modeling. Drake has under development a five course "data-savvy" sequence which will be required of all business students, not just those with a special interest in data analytics.

3. RESEARCH METHODOLOGY

This research uses a "grounded theory" approach. The sociologists Barney Glaser and Anselm Strauss developed grounded theory in the 1960's. In the grounded theory approach, conclusions are drawn and theories are produced by analyzing a body of data. In essence, the theories that are produced are "grounded" in the data (Glaser & Strauss, 1967).

For this project, sixty colleges/universities that offer a business analytics minor were randomly selected. Appendix One lists all of the colleges/universities having a program was included in the study. The curriculum for each of the programs was then reviewed, analyzed, and tabulated. For each program, it was determined the number and nature of the prerequisite courses, required courses, and elective courses. A list of the courses was then recorded.

Once the courses were identified, the researchers then reviewed the catalog descriptions of the courses. Based on the catalog descriptions a list of the topics covered was compiled. The results of that compilation are shown in the following section.

4. RESULTS

The analysis of the sixty programs showed that, on average, business analytics minor programs have two prerequisite courses, three required courses, and two electives. However, there was some variation in this pattern as about a third of the programs had no prerequisite courses and about a quarter of the programs had no elective courses. The maximum number of courses in a program in each of the three categories (prerequisites, required, and electives) was four for prerequisites, seven for required courses, and four for number of electives.

The results of the topics covered in the programs have been compiled into three tables. Table 1 shows a listing of all the prerequisite topics covered by all sixty of the business analytics minor programs that were included in the study. The first column lists the topic covered, the second column shows a count of the number of programs that covered that topic, and the third column shows the percentage of the sixty schools that covered that topic. Table 2 shows a list of all required topics for all sixty schools (again, also showing the count of programs and the percentage of schools). Table 3 shows a list of all electives for all sixty schools in the same format.

The tables are self-explanatory but it is worth noting some observations. First, concerning prerequisite topics, the most popular topics are basic statistics and principles of IT/IS/MIS. It appears that most programs believe that knowledge of these topics is important in preparing students for the required courses. A review of table three supports this, as predictive analytics (the branch of advanced analytics which is used to make predictions about unknown future events) and descriptive analytics (a preliminary stage of data processing that creates a summary of historical data) are the most popular required topics and both rely heavily on a knowledge of basic statistics and of the topics covered in an introductory, principles of IT/IS course.

In addition to predictive and descriptive analytics, Database, Data Mining, Basic Statistics (when not a prerequisite), Decision Science / Business Intelligence, Data Visualization, Excel (Spreadsheets), Principles of IS/IT/MIS (again when not a prerequisite), Introductory Computer Programming, Management Science / Operation Management, and Big Data are also popular required topics. It is also worth noting

Prerequisite Topics	Count	Percent
Basic Statistics	32	53%
Principles of IS/IT/MIS	11	18%
Descriptive Analytics	7	12%
Predictive Analytics	4	7%
Introductory Economics	4	7%
Advanced Statistics	3	5%
Excel (Spreadsheets)	3	5%
Calculus	3	5%
Management Science / Operation Mgt	2	3%
Introductory Computer Programming	2	3%
Prescriptive Analytics	1	2%
Decision Science / Business Intelligence	1	2%
Database	1	2%
Marketing (or Marketing Related)	1	2%
Linear Algebra	1	2%
Intro to Business	1	2%

Table 1 Prerequisite Topics Required By Business Analytics Minor Programs

that seven percent of the programs have a required capstone or project oriented topics course.

As would be expected, based on the emphasis of the individual program, electives offered by the various programs are fairly splintered. The most popular elective topics are Marketing (or Marketing Related), Econometrics, Financial Modeling, Database, Data Mining, Decision Science / Business Intelligence, Simulation and Risk Analysis, Supply Chain Management, Management Science / Operation Management, Forecasting, Application Specific Analytics (Travel, Hospitality, Healthcare, Human Resources, etc.), Data Visualization, Statistical Programming or Software, Experiments and Quality Control, Advanced Statistics, Analytics Research, Project Management, Accounting IS, and Big Data.

Required Topics	Count	Percent
Predictive Analytics	43	72%
Descriptive Analytics	37	62%
Database	27	45%
Data Mining	26	43%
Basic Statistics	22	37%
Decision Science / Business Intelligence	17	28%
Data Visualization	14	23%
Excel (Spreadsheets)	14	23%
Principles of IS/IT/MIS	11	18%
Introductory Computer Programming	9	15%
Management Science / Operation Mgt	7	12%
Big Data	7	12%
Advanced Statistics	6	10%
Capstone/Project	4	7%
Econometrics	4	7%
Simulation and Risk Analysis	4	7%
Prescriptive Analytics	3	5%
Systems Analysis and Design	3	5%
Problem Solving	3	5%
SQL	3	5%
Business Process Analysis	3	5%
Marketing (or Marketing Related)	2	3%
Calculus	2	3%
Financial Modeling	2	3%
Statistical Programming or Software	1	2%
Supply Chain Mgt	1	2%
Project Mgt	1	2%
Forecasting	1	2%
Geographic IS	1	2%
Accounting (Managerial or Financial)	1	2%
Ethics	1	2%
Linear Algebra	1	2%
International or Global Related	1	2%

Table 2 Required Topics Covered By Business Analytics Minor Programs

Elective Topics	Count	Percent
Marketing (or Marketing Related)	28	47%
Econometrics	20	33%
Financial Modeling	15	25%
Database	14	23%
Data Mining	13	22%
Decision Science / Business Intelligence	11	18%
Simulation and Risk Analysis	11	18%
Supply Chain Mgt	10	17%
Management Science / Operation Mgt	9	15%
Forecasting	9	15%
Application Specific Analytics	9	15%
Data Visualization	8	13%
Statistical Programming or Software	8	13%
Experiments and Quality Control	8	13%
Advanced Statistics	7	12%
Analytics Research	7	12%
Project Mgt	7	12%
Accounting IS	7	12%
Big Data	6	10%
Introductory Computer Programming	5	8%
Systems Analysis and Design	5	8%
ERP	5	8%
Web Design & Analytics	5	8%
Prescriptive Analytics	4	7%
Geographic IS	4	7%
Principles of IS/IT/MIS	3	5%
Excel (Spreadsheets)	3	5%
Capstone/Project	3	5%
Predictive Analytics	2	3%
Advanced Economics	2	3%
Accounting (Managerial or Financial)	2	3%
Artificial Intelligence	2	3%
Business Process Analysis	2	3%
Descriptive Analytics	1	2%
Calculus	1	2%
Problem Solving	1	2%
Ethics	1	2%
Cyber Security	1	2%
International or Global Related	1	2%
Text Analytics	1	2%
Managerial Analytics	1	2%

Table 3 Elective Topics Covered By Business Analytics Minor Programs

5. SAMPLE BUSINESS ANALYTICS MINOR CURRICULUM

This section introduces a sample business analytics minor curriculum that was developed based on the results of the data collected. It includes the following courses:

Prerequisites:
1. Basic Statistics
2. Principles of IS/IT/MIS
3. Excel (optional)

Required Courses:
1. Business Analytics I
2. Business Analytics II
3. Management Science

Electives (Choose Two):
1. Marketing or Marketing Related
2. Econometrics
3. Intro to Programming
4. Database Management Systems
5. Decision Support Systems
6. Data Visualization
7. Statistical Programming or Software
8. Application specific course

The role of the prerequisite courses would be to prepare the student for the material covered in the minor. The sample curriculum has two required prerequisites, (Basic Statistics and Principles of IS/IT/MIS), and one optional prerequisite (Excel). The Basic Statistics course would cover statistical theories and techniques commonly used in the analysis of business data. Emphasis is on descriptive measures, probability theory, estimation techniques and forecasting methods, hypothesis testing, and time series analysis. The Principles of IT/IS/MIS course topics would include the following: computer hardware and software architecture, organizing data, telecommunications and networks, types of systems and their development, and the role of information technology in business and society. In the Excel course, students will learn to

navigate Microsoft Excel software and become familiar with Excel's features and capabilities.
Once students have fulfilled the prerequisites, the sample curriculum suggests three required courses: Business Analytics I, Business Analytics II, and Management Science. The Business Analytics I course would provide students with the fundamental concepts and tools needed to understand the emerging role of business analytics in organizations. The course would cover managerial statistical tools in descriptive analytics and predictive analytics, including probability distributions, sampling and estimation, statistical inference, and regression analysis. Students would also learn how to communicate with analytics professionals using basic data visualization techniques to effectively use and interpret analytic models and results for making better business decisions.

The second required course, Business Analytics II, would provide students with advanced concepts and tools needed to understand the role of data analytics in organizations. Topics would include forecasting, risk analysis, simulation, data mining, and decision analysis. Emphasis is on applications, concepts and interpretation of results as well as conducting statistical analyses.

The third required course, Management Science, involves strategic conceptualization, decision-making and analysis of processes within the business and its environment. This course introduces quantitative and computing techniques that contemporary managers use to create models representing the business problems they need to solve. The emphasis of this course will be on the integration and development of modeling skills including problem recognition, data collection, model formulation, analysis, and communicating the results. Building logical thinking and quantitative skills are among the objectives of this course.

The elective courses in the sample curriculum help the students develop skills that increase their knowledge of a specialized area within their field. The sample curriculum suggests two electives. The electives offered are shaped by the goals of the program and, at least to start, the available course offerings at the institution. Based on the results of the analysis, a marketing based course, such as e-commerce or e-marketing, and an econometrics course should be included in the electives offered.

For instance, an e-marketing course could examine how analytics, the Internet, and related technologies are transforming the ways in which firms market their products and services. Topics covered would include examination of emerging business models, application of analytics, customer relationship management (CRM), role of data mining and data warehouses, personalization, branding issues, dynamic pricing and price competition, role of the Internet as a communications / advertising medium, distribution through the Internet and e-tailing, and legal and ethical issues.

Econometrics is the application of mathematical and statistical methods and techniques in order to: 1) help understand, analyze, and interpret economic and financial data, 2) test economic and financial hypotheses/theories, and 3) generate predictions about particular economic and financial variables. Econometrics is fundamentally a regression-based correlation methodology used to measure the overall strength, direction, and statistical significance between a "dependent" variable - the variable whose movement or change is to be explained - and one or more "independent" variables that will explain the movement or change in the dependent variable.

Technology based courses are also popular electives in our research pool. These courses help to define the utilization of analytical tools and have the added benefit of most likely already existing in the catalog of IS courses within the business school. For instance, Intro to Programming, Database Management Systems, Decision Support Systems, Data Visualization (any effort to help people understand the significance of data by placing it in a visual context), Statistical Programming or Statistical Software are all courses that help the student understand technologies important to the analytics process.

One last note on electives. In order to tailor the minor program to the student’s major, an application specific course is suggested. This course would be housed, and staffed, by the department of the major. For instance, if a student mas majoring in Health Administration, a course called “Analytics for Health Administration” could be offered by the Health Administration department yet count as an elective in the business analytics minor program. Again, this course could come from any major such as Accounting, Travel, Hospitality, Healthcare, Human Resources, etc.

6. CONCLUSION

Given the high demand for business analytics skills and the ubiquitous nature of the analytics field, it is an ideal candidate as an academic minor. This research analyzed the curricula of sixty current business analytics minor programs in order to answer the question as to what courses and topics should be included in the curriculum.

The analysis showed that the ideal business analytics minor would consist of prerequisite courses in statistics and IT/IS principles, and required courses that cover introductory analytics (descriptive, predictive, regression, probability, visualization, etc.) and advanced analytics topics such as forecasting, risk analysis, simulation, data mining, and decision analysis. In addition, a required course in Management Science should be included in the curriculum. Furthermore, this research suggests that the BA minor should include elective courses designed to tailor the program to the individual goals of the student.

7. REFERENCES

Aasheim C. L., Rutner, Williams, & Gardiner (2014). Big Data Analytics and Data Science Undergraduate Degree Programs. *2014 Annual Meeting of the Decision Sciences Institute Proceedings* p. 338 - 359 Available at: http://works.bepress.com/adrian-gardiner/8/

Computerworld.com (2018). http://www.computerworld.com Accessed June 2, 2018

Glaser, B. G. & Strauss, A. L. (1967). The Discovery of Grounded Theory: Strategies for Qualitative Research, Aldine Publishing Company, Chicago, IL.

Gorman, M. and Klimberg, R. (2014). Benchmarking Academic Programs in Business Analytics. Interfaces. 44(3), 253-350.

Mckinsey.com (2018). Big data: The next frontier for innovation, competition, and productivity. https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/big-data-the-next-frontier-for-innovation. Accessed May 28, 2018

Meyer, B. (2015). The Age of Data Hits the University Curriculum: The Development of a Cross-College Data Analytics Program in Decision Sciences in the 21st Century: Theoretical Impact and Practical Relevance. Proceedings of the 2015 Annual Meeting of the Decision Sciences Institute, Seattle, Washington, 2015. Available at https://decisionsciences.org/publications/annual-meeting-proceedings/2015-annual-meeting-proceedings/

Phelps, A. & Szabat, K. (2015). Business Analytics and Data Science at Higher Education Institutions: Who is Teaching What? Joint Statistical Meetings, Seattle, Washington, 2015. Available at http://www.statlit.org/pdf/2015-Phelps-Szabat-ASA-Slides.pdf.

Rostcheck, D. (2018). Data Science? Business intelligence? What's the Difference? Linked in. https://www.linkedin.com/pulse/data-science-business-intelligence-whats-difference-david-rostcheck/ Accessed September 10, 2018.

U.S. Bureau of Labor Statistics (2018). www.bls.gov. Accessed May 15, 2018.

Wang, Y. (2015). Business Intelligence and Analytics Education: Hermeneutic Literature Review and Future Directions in IS Education. Proceedings of the Twenty-First Americas Conference on Information Systems (AMCIS), Puerto Rico, 2015. Available at SRRN: https://ssrn.com/abstract=2603365

Watson, H. (2013). The business case for analytics. BIZED XII(3), 48-54.

Wilder, C. & Ozgur, C. (2015). Business Analytics Curriculum for Undergraduate Majors. INFORMS: Transactions on Education, 15(2), 180-187.

Wymbs, C. (2016). Managing the Innovation Process: Infusing Data Analytics into the Undergraduate Business Curriculum (Lessons Learned and Next Steps). Journal of Information Systems Education, 27(1), 61-74.

Zheng, J. (2018). Business Intelligence and Analytics: A Comprehensive Overview. Available at http://jackzheng.net/teaching/it6713/

APPENDIX ONE – A LIST OF INSTITUTIONS INCLUDED IN THE STUDY

Ashland University
Auburn University
Baruch College
Drexel University
Eastern Illinois University
Fairfield University
Florida Atlantic University
Florida State University
Florida International University
Georgia Southern University
Hamline University
James Madison University
Kent State University
LeMoyne College
Loyola University
Manhattan College
Miami University (Ohio)
Northeastern University
Northern Illinois University
Northern Kentucky University
Northwood University
Ohio University
Old Dominion
Pace University
Penn College
Purdue University
Rider University
Sabanci University
Saint Joseph's University
Santa Clara University

Southern Methodist University
SUNY Plattsburgh
Temple University
Texas A&M University
University of Arkansas
University of Central Missouri
University of Cincinnati
University of Connecticut
University of Delaware
University of Denver
University of Hartford
University of Idaho
University of Illinois at Chicago
University of Maryland
University of Miami
University of Minnesota
University of Nebraska–Lincoln
University of San Diego
University of San Francisco
University of Scranton
University of South Dakota
University of Tampa
University of Tulsa
University of Wisconsin
Valparaiso University
Villanova University
Western Washington University
Wilkes University
Xavier University
York College of Pennsylvania

Hour of Code: A Study of Gender Differences in Computing

Jie Du
dujie@gvsu.edu
Grand Valley State University
Allendale, MI 49401

Hayden Wimmer
hwimmer@georgiasouthern.edu
Georgia Southern University
Statesboro, GA 30460

Abstract

Computer programmers in the U.S. labor force are facing a shortage. Focusing on recruiting females has the potential to address this shortage. Computing is a male dominated field which provides an opportunity to recruit the other 50% of the population, females, to fill the open positions. This work studies gender differences in computer programming based on an Hour of Code tutorial. Following a pre- and post-test design, this work demonstrates that males have significantly more previous exposure to computer programming and are significantly more interested in pursuing computer programming. Results also indicate that females do equally as well or better in programming comprehension. In one comprehension question following the tutorial, women significantly outperformed men demonstrating that women may have a higher aptitude for computer programming; however, they are underrepresented in the job market. Based on our results, we suggest that more should be done in early formative years to attract females into computer programming to aid in filling the gap of the projected employment market.

Keywords: Gender, computer science, hour of code, programming, and survey.

1. INTRODUCTION

Computing is one of the fasting growing industries; however, many positions go unfilled due to a lack of qualified individuals. Males dominate STEM (Science, Technology, Engineering and Mathematics) related fields with computing being no exception. Efforts to decrease the gender gap in computing has brought efforts such as the Women in IT program (WIT) as well as programs to recruit females at a younger age, such as Girls Who Code (GWC). While the aforementioned efforts intend to make progress, GWC founder and CEO states: “That was by far the most surprising thing—it's only getting worse, It feels like today computer science is becoming more popularized and it's true that the pool is getting bigger, but the share of women has declined."(Zarya, 2016).

Research has been focused on the question of why we are losing women in the Computer Science (CS) field for decades. The dominant framework that seeks to answer this question is usually condensed through the metaphor of a ‘leaky pipeline’ (Vitores & Gil-Juárez, 2016), which describes how women drop out of STEM fields at all stages of their careers (Soe & Yakura, 2008). In order to close the gender gap, a first, necessary step is to gain an understanding of gender differences in computing. We posit that females are as competent in computing; however, efforts to recruit females to computing are not adequate. To investigate our assertions, we employ the widely popular code.org. We have students take a pre- and post-test to measure interest in taking a programming course and their knowledge of programming basics, such as programming structures.

Results indicate that males had more previous exposure to computer programming. Furthering this, following the pre- and post-test, males are more likely to enroll in a programming course. In one basic programming comprehension question (loop), women performed better than men indicating women have improved performance following the hour of code tutorial. Based on our results, we recommend that more effort needs to be undertaken to recruit females into computer programming by targeting them at a younger age.

The remainder of this paper is structured as follows: first we present relevant literature in section 2, section 3 details our methodology, section 4 illustrates our results, section 5 discusses the implications of our results, and section 6 concludes this work.

2. LITERATURE REVIEW

Gender Issues in Computer Science

The field of CS has been defined as the 'incredible shrinking pipeline' (Camp, 1997). Soe and Yakura (2008) found that at each stage, the pipeline 'leaks' more for women than it does for men. The leaking problem is worse at the high school level as the field continues to lose the participation and interest of a broad layer of students, especially females (Goode, Estrella, & Margolis, 2013). Gender differentials, school/family influences, and stereotyping of science are the three main contributors to gender gap in the STEM fields (Acker, 1987).

Gender plays an important role in decisions about the choice of one's major and ultimately one's profession (Beyer, 1999). Males report more comfort and confidence with computers than do females (Temple & Lips, 1989). Males show a more positive attitude toward computers than do females even when computer experience is controlled (Kadijevich, 2000). Another interesting finding is that at younger ages, there was no difference between boys and girls in using computer but however the interest level of the girls diminished at later stages (Calvert, 2005). Females are less attracted to formal CS education than males (Shashaani, 1997). The top reason to choose a CS major for women was their desire to use it in another field while for men was their interest in computer games (Carter, 2006). Women who earned less than B in CS courses were more likely to quit a CS major, and men who earned less than B were more likely to continue taking CS courses (Katz, Allbritton, Aronis, Wilson, & Soffa, 2006). Females have a higher level of computer anxiety which reduces their self-effectiveness which in turn leads to increased perceptions of the effort required to use IT (Venkatesh & Morris, 2000).

Stereotypes based on gender widely exist in CS. One of the most well-known stereotypes is the low awareness of female academic competence (Koch, Müller, & Sieverding, 2008). However, research has shown that gender stereotypes in the academic domain are often inaccurate (Beyer, 1999). Beyer surveyed nearly 300 college students and found out that despite higher GPAs by females in masculine majors, participants believed that males have higher GPAs. Female students outperformed males with respect to academic achievements at both the high school and college levels (Fan & Li, 2005). A study of an introductionary CS couse found that women who reported having less experience of programming skills outperformed men who reported having a high level of programming experience (Kadijevich, 2000). Women reported more stereotype-consistent perceptions than did men (Ehrlinger et al., 2017).

A lot of research focuses on ways to remedy the gender disparities. A good start is to increase women's awareness of and experience in CS when they are young (Kermarrec, 2014). A significant correlation between early prior computing experiences and success by females in a college computer course was detected in (Taylor & Mounfield, 1994). Therefore, outreaching girls to get them in contact with computers can reduce gender differences in computer attitudes (Sáinz & López-Sáez, 2010). Outreach efforts should focus on ways to engage parents because the influence of family is found to play a critical role in encouragement and exposure (Wang, Hong, Ravitz, & Ivory, 2015).

Vilner and Zur (2006) found that women had difficulties in passing the courses during the first stages of the curriculum and not at the later stages. This finding suggests helping women succeed in their first CS courses will retain them in CS. Also, using virtual environments to communicate a sense of belonging among women can help attract and retain more women in CS (Cheryan, Meltzoff, & Kim, 2011).

Hour of Code

Code.org was launched in 2013 as a nonprofit dedicated to promoting CS education and increasing participation by women and underrepresented minorities. Code.org provides a curriculum for K-12 computer science and the majority of the students who took those courses are girls or underrepresented minorities.

The term “hour of code” refers to an hour introduction to CS. Initiated by code.org, hour of code began as an hour coding tutorial to show students that programming is fun and creative. Nowadays, hour of code has developed as a global movement breaking stereotypes to encourage kids to learn CS. Hourofcode.com offers 100+, one-hour-long, computer science activities. Those activties are online and work with computers or mobile devices.

A study was conducted online over the course of five days in December 2016 as part of Computer Science Education Week (Phillips & Brooks1, 2017). The findings suggest that hour of code impacts student attitude toward and self-efficacy with CS positively, especially for females in K-12. Code.org advocates that “an Hour of Code is a great place to start addressing the diversity gap and introducing computer science to more girls in an engaging and empowering way!”(Code.org, 2018).

We aim to investigate the impact of hour of code on college student attitude toward programming and their knowledge of programming based on gender.

3. METHODOLOGY

Data Collection
This study attempts to gain insights on the gender difference on attitude toward programing, computing skills, and experiences related to programming by surveying students in an introductory computing course at a public university. An electronic Likert-scale questionnaire was implemented to survey the subjects. The 14 survey questions are based on a Likert scale and from (Du, Wimmer, & Rada, 2016) (see Appendix).

This study has three steps:

- *Step 1:* The participants were asked to complete a pre-survey before taking an hour of code tutorial. The pre-survey contains Q1 to Q11 plus Q14.
- *Step 2:* The participants were asked to take the tutorial “Write Your First Computer Program” from the category of “Tutorial for Beginners.” This tutorial was selected because most of our participants are first-year college students and they have very limited programming experience.
- *Step 3:* The participants were asked to take a post-survey when they finished the tutorial. The post-survey contains Q1, Q6 to Q13. Participants’ responses to the pre- and post-surveys were matched using a PIN number created by each participant (Q1).

One hundred and eleven students who have enrolled in an introductory computing course during the winter semesters in 2017 and 2018 participated this study. Q11 contains 22 missing responses and thus is removed from our data analysis. Besides Q11, Q12 and Q13 contain one missing value, respectively from one participant. Accordingly, that participant’s responses were removed from the data set. Therefore, after removing responses that are unmatched or contain missing values, the data collection yielded 99 pairs of useable surveys (48 pairs in 2017 and 51 pairs in 2018). Table 1 summaries the demographics of the sample (Q2 to Q4). Table 2 shows the descriptive statistics for the main questions (Q5 to Q13).

Demographic	Category	Percentage
Age (Q2)	<19	27.3
	19-22	66.7
	22-26	6.0
	>26	0
Gender (Q3)	Male	61.6
	Female	38.4
Major (Q4)	Business	64.6
	Non-business	35.4

Table 1. Demographics of Participants

Question	Min	Max	Mean	Std. Dev
Q5	1	2	1.75	0.437
Q6	1	3	1.08	0.340
Q7_ Pre	1	5	3.16	1.066
Q8_ Pre	1	4	2.16	1.017
Q9_ Pre	0	1	0.74	0.442
Q10_ Pre	0	1	0.43	0.498
Q7_ Post	1	5	3.47	1.063
Q8_ Post	1	4	2.44	0.992
Q9 _Post	0	1	0.77	0.424
Q10_ Post	0	1	0.46	0.501
Q12	1	5	3.88	0.872
Q13	1	5	3.55	0.824

Table 2. Descriptive Statistics of Items

Data Analysis
We look at the gender difference about programming by splitting the whole data file into two sub datasets: 61 male students vs. 38 female students. We conducted a three-step analysis to examine the gender difference on programming. The three steps are:

1. We look at the data in the **PRE**-survey to detect any difference between females and males regarding their attitude toward programming and their understanding on basic programming ideas.
2. The same analysis is then conducted on the **POST**-survey dataset.
3. We compare the changes between pre- and post-surveys on both females and males.

4. RESULTS

Pre-Survey Data Analysis
An independent sample *t*-test was conducted on the pre-survey data between male and female students (see Table 3). We found that two significant differences between females and males: males report having taken more programming courses than do females; males are more likely to take a programming course than are females.

While not statistically significant, it is interesting to find that the female students had a higher average accuracy rate when answering the two programming comprehension questions (Q9 and Q10) than did male students. Future work will more closely examine this relationship. Table 4 reports the accuracy rate on the two questions in both groups.

Items	t	Sig. (2-tailed)	Mean Diff.	Std. Err. Diff.
Q5	-1.814	.073*	-.154	.085
Q6	1.499	.137	.088	.059
Q7_Pre	.027	.978	.006	.221
Q8_Pre	2.205	.030**	.433	.196
Q9_Pre	-.456	.649	-.042	.092
Q10_Pre	-1.458	.148	-.149	.102

*p<0.1, **p<0.05
Table 3. T-test for Pre-Survey Data

Items	Male	Female
Q9_Pre	72.1%	76.3%
Q10_Pre	37.7%	52.6%

Table 4. Accuracy Rates in Pre-Survey

Post-Survey Data Analysis
Another independent sample *t*-test was conducted on the post survey data between the male and female students (see Table 5). The *t*-test results show that significant differences exist between males and females on two questions (Q8 and Q10). We found that males are more likely to take a programming course following an hour of code tutorial than females. Females outperformed males on one comprehension question. Table 6 reports the accuracy rates on the two programming comprehension questions in both groups after the subjects completed an hour of code tutorial.

Items	t	Sig. (2-tailed)	Mean Diff.	Std. Err. Diff.
Q7_Post	.008	.994	.002	.221
Q8_Post	2.545	.013**	.508	.200
Q9_Post	.569	.571	.050	.088
Q10_Post	-1.811	.073*	-.186	.102
Q12	-.347	.730	-.069	.198
Q13	.682	.497	.116	.171

*p<0.1, **p<0.05
Table 5. T-test on Post-Survey Data

Question	Male	Female
Q9_Post	78.7%	73.7%
Q10_Post	39.3%	57.9%

Table 6. Accuracy Rates in Post-Survey

For the two questions that only appeared in the post-survey, Q12 and Q13, most of the participants offered very positive responses. 76% of the female students and 84% of the male students enjoyed the hour of code tutorial (see Figure 1). 53% of the female students and 62% of the male students expressed that the hour of code changed their attitude toward programming positively (see Figure 2). It seems that the male students are more responsive to the game-enhanced tutorial than their female peers.

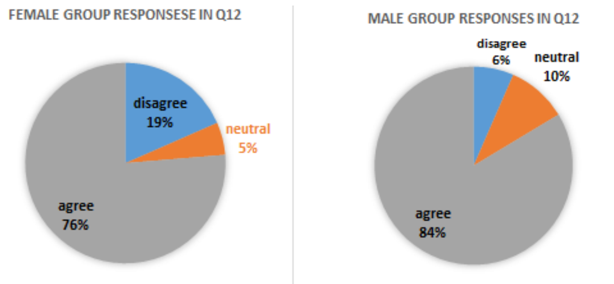


Figure 1. Responses to Q12

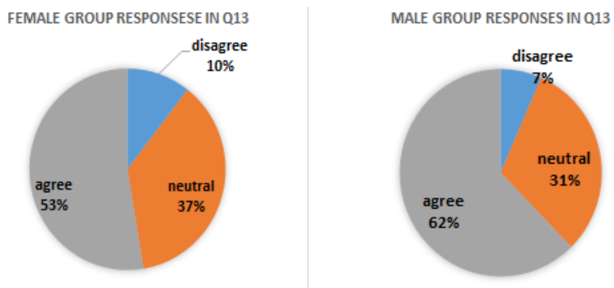


Figure 2. Responses to Q13

Pre-Survey Data vs. Post-Survey Data

A paired sample *t*-test was conducted to examine the changes in the four questions that were asked in both pre- and post-surveys (see Table 7 and 8). Two questions (Q7 and Q8) received significantly different responses in the pre- and post-surveys. The *t*-test results show that the hour of code tutorial has significantly changed students' attitude toward programming. No significant difference based on gender was detected. After taking the hour of code, the students appreciate the importance of learning programming better and are more willing to take programming courses (see Figures 3 -6).

Female Group		Paired Differences: Mean	Std. Dev.	Std. Error Mean	Confidence: Lower	Upper	t	df	Sig. (2-tailed)
Pair 1	Q7_Pre - Q7_Post	-.316	.904	.147	-.613	-.019	-2.154	37	.038
Pair 2	Q8_Pre - Q8_Post	-.237	.634	.103	-.445	-.028	-2.303	37	.027
Pair 3	Q9_Pre - Q9_Post	.026	.545	.088	-.153	.205	.298	37	.767
Pair 4	Q10_Pre - Q10_Post	-.053	.517	.084	-.223	.117	-.627	37	.534

Table 7. T-test on the Female Participants Before vs After Taking the Hour of Code Tutorial

Male Group		Paired Differences: Mean	Std. Deviation	Std. Error Mean	Interval of the: Lower	Upper	t	df	Sig. (2-tailed)
Pair 1	Q7_Pre - Q7_Post	-.311	.672	.086	-.484	-.139	-3.621	60	.001
Pair 2	Q8_Pre - Q8_Post	-.311	.534	.068	-.448	-.175	-4.559	60	.000
Pair 3	Q9_Pre - Q9_Post	-.066	.442	.057	-.179	.048	-1.158	60	.251
Pair 4	Q10_Pre - Q10_Post	-.016	.387	.050	-.115	.083	-.331	60	.742

Table 8. T-test on the Male Participants Before vs After Taking the Hour of Code Tutorial

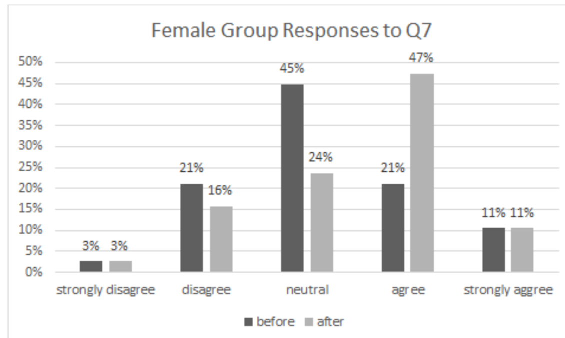


Figure 3. Female Students' Responses to Q7

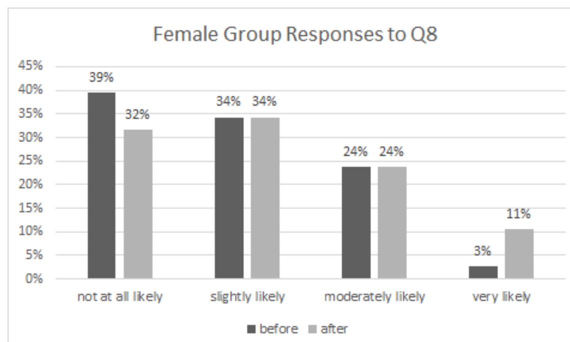


Figure 4. Female Students' Responses to Q8

After taking the hour of code, male participants appreciate more the importance of learning programming than females. Males are more willing to take programming courses than females.

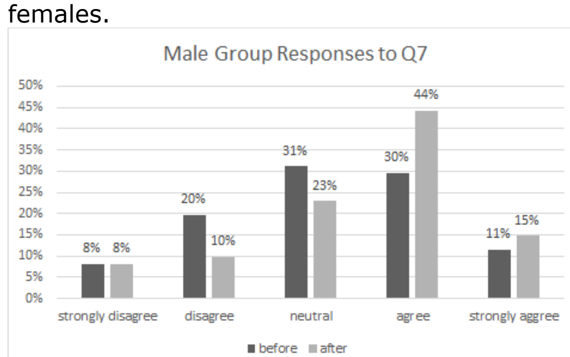


Figure 5. Male Students' Responses to Q7

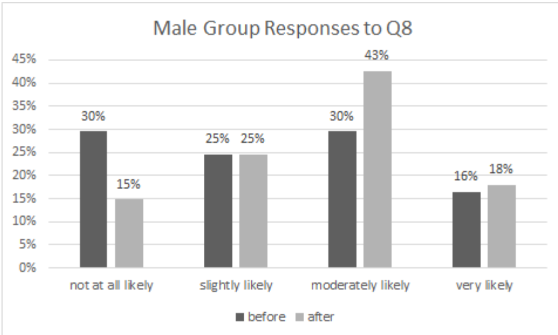


Figure 6. Male Students' Responses to Q8

5. DISCUSSIONS

In this section, we discuss our findings, the limitations and implications of our study, and highlight future directions.

Findings

Our study found that the hour of code tutorial significantly changed the participants' attitude toward programming and there is no gender difference on the changes.

Before taking the hour of code tutorial, more male participants (41%) believe that programming is very important compared to their female peers (32%). More male participants (46%) are likely to take a programming course compared to their female peers (27%).

After taking the hour of code tutorial, 58% of the female participants and 59% of the male participants agree that everyone should learn coding. More male participants (61%) are likely to take a programming course compared to their female peers (35%). But we did not find these differences were statistically significant.

It is interesting to find that the hour of code improves student knowledge about programming differently based on gender. While not statistically significant, before taking hour of code, the accuracy rates in the female group are higher than the male group. A significant difference was detected on one programming comprehension question (Q10) between the two gender groups after the subjects completed an hour of code tutorial. The female students had a significantly better accuracy rate than did the male students. The results suggest that following the tutorial women scored significantly higher than men in comprehending one programming structure, loop.

Implications

In this subsection, two key topics are addressed: whether females are capable to learning programming and how to motivate females to learn programming.

Females are doing better than their male peers learning programming

A stereotype exists that boys are born to be good at computing while girls are born to be good at other fields not related to computing. Our findings suggest that this belief might be a faulty perception. Females are doing better than males on the coding questions specifically asked in our survey. This should give female students more confidence and encouragement when they are introduced to the computing field.

Ways to motivate female students to learn coding

We found that our female participants are more reluctant to taking a programming course compared to their male peers. This indicates that female students need more motivation and encouragement when introduced to the computing field. Although female students are capable of coding, they still are reluctant to try it. Our study is only the first step to help female students gain confidence on computing. How to help them to fight the gender stereotype becomes a very important topic that needs to be addressed by educators who strive to close the gender gap. Some factors including interest, confidence that they can succeed in this career, feeling like they belong with others in this occupation, and identifying themselves as this "type of person" are believed to play an important role in recruiting girls to try computer science (ncwit.org, 2018).

Limitations and Future Directions

The major limitation of our study is the limited sample size. A much larger sample would give more reliable statistical results. This study serves as a pilot study and highlights several interesting points that deserve further investigations. It is important to outreach young girls in K-12 to introduce the concept of computing and this helps boost girls' self-efficacy on computing.

In our current study, we use a tutorial called "Write Your First Computer Program". This tutorial invited the student to work through 20 progressively more complex puzzles. Those puzzles were designed based on some popular video games (e.g. Angry Bird) or movies (e.g. Ice Age). Educators and instructors could explore the hour of code tutorials offered at code.org to identify other tutorials that might work better for girls. The effectiveness of utilizing Hour of Code to promote CS enrollment could be further evaluated based on students' actual enrollment behaviors after they express interest by doing Hour of Code tutorials.

6. CONCLUSIONS

Currently and projected into the future, there are a shortage of computer programmers in the United States. Computer programming is largely male dominated leaving females underrepresented even though they represent approximately 50% of the population. This work shows that males had significantly higher exposure to computer programming prior to enrolling in a college level introduction to computing course. Males were also more inclined to enroll in a computer programming course following the hour of code tutorial. Interestingly, females either equaled or outperformed males on programming comprehension questions following the tutorial. Based on this, we recommend more interactive activities to boost girls' self-efficacy and sense of belonging to the computing field need to be done in early and formative years to recruit females into computer programming. Recruiting females has the potential to increase the computer programming workforce and fill the projected shortages of computer programmers.

7. REFERENCES

Acker, S. (1987). Feminist theory and the study of gender and education. *International Review of Education, 33*, 419-435.

Beyer, S. (1999). The Accuracy of Academic Gender Stereotypes. [journal article]. *Sex Roles, 40*(9), 787-813. doi: 10.1023/a:1018864803330

Calvert, S. (2005). *Age, Ethnicity, and Socioeconomic Patterns in Early Computer Use: A National Survey* (Vol. 48).

Camp, T. (1997). The incredible shrinking pipeline. *Commun. ACM, 40*(10), 103-110. doi: 10.1145/262793.262813

Carter, L. (2006). Why students with an apparent aptitude for computer science don't choose to major in computer science. *SIGCSE Bull., 38*(1), 27-31. doi: 10.1145/1124706.1121352

Cheryan, S., Meltzoff, A. N., & Kim, S. (2011). Classrooms matter: The design of virtual classrooms influences gender disparities in computer science classes. *Computers & Education, 57*(2), 1825-1835. doi: http://dx.doi.org/10.1016/j.compedu.2011.02.004

Code.org. (2018). The largest learning event in history Retrieved June 12, 2018 from https://code.org/files/HourOfCodeImpactStudy_Jan2017.pdf

Du, J., Wimmer, H., & Rada, R. (2016). "Hour of Code": Can It Change Students' Attitudes toward Programming? *Journal of Information Technology Education: Innovations in Practice, 15*, 53-73. doi: 10.28945/3421

Ehrlinger, J., Plant, E. A., Hartwig, M. K., Vossen, J. J., Columb, C. J., & Brewer, L. E. (2017). Do Gender Differences in Perceived Prototypical Computer Scientists and Engineers Contribute to Gender Gaps in Computer Science and Engineering? *Sex Roles*, 1-12.

Fan, T.-S., & Li, Y.-C. (2005). Gender issues and computers: college computer science education in Taiwan. *Computers & Education, 44*(3), 285-300. doi: http://dx.doi.org/10.1016/j.compedu.2004.02.003

Goode, J., Estrella, R., & Margolis, J. (2013). Lost in Translation: Gender and High School Computer Science. In J. Cohoon & W. Aspray (Eds.), *Women and Information Technology: Research on Underrepresentation*. Cambridge, MA: MIT Press.

Kadijevich, D. (2000). Gender Differences in Computer Attitude among Ninth-Grade Students. *Journal of Educational Computing Research, 22*(2), 145-154.

Katz, S., Allbritton, D., Aronis, J., Wilson, C., & Soffa, M. L. (2006). Gender, achievement, and persistence in an undergraduate computer science program. *Database for Advances in Information Systems, 37*(4), 42-57.

Kermarrec, A. M. (2014). Computer Science: Too Young to Fall into the Gender Gap. *IEEE Internet Computing, 18*(3), 4-6. doi: 10.1109/mic.2014.48

Koch, S., Müller, S., & Sieverding, M. (2008). Women and computers. Effects of stereotype threat on attribution of failure. *Comput. Educ., 51*(4), 1795-1803. doi: 10.1016/j.compedu.2008.05.007

ncwit.org. (2018). Top 10 Ways of Recruiting High School Women into Your Computing Classes Retrieved July 3, 2018 from https://www.ncwit.org/resources/top-10-ways-recruiting-high-school-women-your-computing-classes/top-10-ways-recruiting

Phillips, R. S., & Brooks1, B. P. (2017). The Hour of Code: Impact on Attitudes Towards and Self-Efficacy with Computer Science Retrieved June 12, 2018 from https://code.org/files/HourOfCodeImpactStudy_Jan2017.pdf

Sáinz, M., & López-Sáez, M. (2010). Gender differences in computer attitudes and the choice of technology-related occupations in a sample of secondary students in Spain. *Computers & Education, 54*(2), 578-587. doi: http://dx.doi.org/10.1016/j.compedu.2009.09.007

Shashaani, L. (1997). Gender Differences in Computer Attitudes and Use among College Students. *Journal of Educational Computing Research, 16*(1), 37-51. doi: https://doi-org.ezproxy.gvsu.edu/10.2190/Y8U7-AMMA-WQUT-R512

Soe, L., & Yakura, E. K. (2008). What's Wrong with the Pipeline? Assumptions about Gender and Culture in IT Work. *Women's Studies, 37*(3), 176-201. doi: 10.1080/00497870801917028

Taylor, H. G., & Mounfield, L. C. (1994). Exploration of the Relationship between Prior Computing Experience and Gender on Success in College Computer Science. *Journal of Educational Computing Research, 11*(4), 291-306. doi: https://doi-org.ezproxy.gvsu.edu/10.2190/4U0A-36XP-EU5K-H4KV

Temple, L., & Lips, H. M. (1989). Gender differences and similarities in attitudes toward computers. *Computers in Human Behavior, 5*(4), 215-226. doi: http://dx.doi.org/10.1016/0747-5632(89)90001-0

Venkatesh, V., & Morris, M. G. (2000). Why don't men ever stop to ask for directions? Gender, social influence, and their role in technology acceptance and usage behavior. *MIS Q., 24*(1), 115-139. doi: 10.2307/3250981

Vilner, T., & Zur, E. (2006). Once she makes it, she is there: gender differences in computer science study. *SIGCSE Bull., 38*(3), 227-231. doi: 10.1145/1140123.1140185

Vitores, A., & Gil-Juárez, A. (2016). The trouble with 'women in computing': a critical examination of the deployment of research on the gender gap in computer science. *Journal of Gender Studies, 25*(6), 666-680. doi: 10.1080/09589236.2015.1087309

Wang, J., Hong, H., Ravitz, J., & Ivory, M. (2015). *Gender Differences in Factors Influencing Pursuit of Computer Science and Related Fields*. Paper presented at the Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education, Vilnius, Lithuania.

Zarya, V. (2016). Here's How to Triple the Number of Girls Who Code Retrieved June 14, 2018 from http://fortune.com/2016/10/20/gender-gap-coding-accenture/

APPENDIX

<table>
<tr><td>Categories</td><td>Questions</td></tr>
<tr><td>PIN</td><td>Q1: Create a PIN and use it in the pre- and post-surveys</td></tr>
<tr><td>Age</td><td>Q2: What is your age?
(under 19, 19-21, 22-26, over 26)</td></tr>
<tr><td>Gender</td><td>Q3: Gender (with which you identify most)?
(Female/Male)</td></tr>
<tr><td>Major</td><td>Q4: What is your major?
(Self-reported)</td></tr>
<tr><td rowspan="2">Prior Experience</td><td>Q5: Have you ever taken any programming courses?
(Yes/No)</td></tr>
<tr><td>Q6: What's your experience with programming?
(Less than 1 year, 2-3 years, 4-5 years, and 5+ years)</td></tr>
<tr><td rowspan="2">Attitude toward Programming</td><td>Q7: To what extent do you agree or disagree with the following statement: Everybody in this country should learn how to program a computer because it teaches you how to think. (disagree/agree)</td></tr>
<tr><td>Q8: How likely are you to take a programming course? (not likely/likely)</td></tr>
<tr><td rowspan="3">Programming Comprehension</td><td>Q9: Which of the lettered choices is equivalent to the following decision?
if x> 10 then
    if x>y then
        Print “x”
    endif
endif
a. If x>10 or y>10 then print “x”
b. If x>10 and x>y then print “x”
c. If y>x then print “x”
d. If x>10 and y>10 then print “x”</td></tr>
<tr><td>Q10: In the following pseudocode, what is printed?
g = 6
h = 4
while g < h
    g = g+1
endwhile
print g, h
a. nothing
b. 4 6
c. 5 6
d. 6 4</td></tr>
<tr><td>Q11: In the following pseudocode, what is printed?
a = 1
b = 2
c=a
a=b
b=c
print a, b</td></tr>
</table>

	a. nothing b. 1 2 c. 2 1 d. None of the above
Comments on Hour of Code	Q12: Did you enjoy the tutorial provided by code.org? (disagree/agree)
	Q13: Completing the tutorial changed your attitude towards programming how? (negative/positive)
Additional Comments	Q14: In the space below, please share any additional comments regarding programming