INFORMATION SYSTEMS EDUCATION JOURNAL

Volume 21, No. 1
March 2023
ISSN: 1545-679X

In this issue:

We have bundled together a number of papers on the general theme of information systems program offerings, including a look at autonomous vehicles in IS education, micro-credentials, user experience design in community projects, first-year experience courses, exam complexity, and IS2020 across the curriculum.

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 five times per year. The first year of publication was 2003.

ISEDJ is published online (https://isedj.org). Our sister publication, the Proceedings of EDSIGCON (https://proc.iscap.info) features all papers, abstracts, 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 ISCAP conference. All papers, whether award-winners or not, are invited to resubmit for journal consideration after applying feedback from the Conference presentation. Award winning papers are assured of a publication slot; however, all re-submitted papers including award winners are subjected to a second round of three blind peer reviews to improve quality and make final accept/reject decisions. 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 36%.

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

INFORMATION SYSTEMS EDUCATION JOURNAL

Micro-Credentials in US Higher Education: An Empirical Analysis

Guido Lang
guido.lang@quinnipiac.edu
Business Analytics & Information Systems
Quinnipiac University
Hamden, CT 06518

Jason H. Sharp
jason.sharp@uvu.edu
Information Systems & Technology
Utah Valley University
Orem, UT 84058

Abstract

Micro-credentials have received renewed attention by universities as a potential pathway to high-demand careers. The goals of this study were to understand the prevalence of micro-credentials offered by universities, the characteristics of universities offering micro-credentials generally and information systems (IS) micro-credentials specifically, and the content of IS micro-credentials. To this end, this study found that only very few universities are currently offering micro-credentials. Specifically, of all 1,860 universities listed on US News & World Report, only 127 (6.8%) offer micro-credentials on Credly, which is one of the largest digital credentialing platforms. However, once universities offer micro-credentials, they do so prolifically. In fact, the 127 universities offer a total of 2,308 micro-credentials, with 114 (89.9%) universities offering more than one micro-credential for an average of 53 micro-credentials offered per university. Moreover, this study found that universities offering micro-credentials or IS micro-credentials are more likely to be tier 1, public universities located in a city, urban, or suburban setting with a higher starting salary of graduates. Lastly, of the 2,308 micro-credentials offered by universities, only 195 (8.4%) are IS micro-credentials addressing competency realms defined in the IS2020 curriculum. The top IS2020 competency realms are technology, data, and development. The most frequently associated skills among IS micro-credentials are cybersecurity, Excel, AI/ML, SQL and HTML, CSS, JS. Future research is needed to understand the longitudinal development of micro-credential offerings among universities (including international universities), on various digital credentialing platforms, possibly leveraging automatic data collection and analysis methods.

Keywords: micro-credentials, digital badges, IS2020

1. INTRODUCTION

While the concept and implementation of digital badges and micro-credentials is not a new phenomenon (e.g., Abramovich, Schunn, & Higashi, 2013; Ahn, Pellicone, & Butler, 2014; Beattie, 2014; Carey, 2012; Rubleske & Cata, 2017), with the reported lack of qualified future computing professionals and the challenges and opportunities afforded by the ongoing COVID-19 pandemic, micro-credentials and digital badges are receiving renewed attention. This is evidenced by a recent call for grant proposals from the Texas Higher Education Coordinating

Board entitled, "Accelerating Credentials of Purpose and Value Grant Program" (THECB, 2022), the purpose of which is:

> the development and/or expansion of the development of short-term postsecondary credentials that incorporate skills and knowledge required by high-demand careers in three broad areas: digital skills, including programming, web and application development, digital project management and cybersecurity programs; data analytics including data analysis and visualization; and front-line healthcare including nursing, medical specialist and technician programs (THECB, 2022, 1).

Further evidence is the growing number of colleges and universities that are now offering digital badges and micro-credentials along with their traditional degree programs. Digital badges "signify accomplishments such as completion of a project, mastery of a skill, or marks of experience" (Casilli & Knight, 2012, p. 1), while a micro-credential represents a verifiable set of competencies in a particular skill or area of expertise (Casilli & Hickey, 2016; Fanfarelli & McDaniel, 2019). Much like the prediction of early adopters that Massive Open Online Courses would revolutionize higher education and take the place of traditional universities and colleges (Ayala, Dick, & Treadway, 2014), early promoters of digital badges and micro-credentials envisioned a similar revolution (Crossley, 2021). While that revolution may not have fully materialized, as institutions of higher education seek to gain competitive advantage and remain relevant in a society becoming ever increasingly hostile toward the value of a four-year college education, micro-credentials and badges have grown increasingly more popular and appear to be a continuing wave of the future. Such that, the number of higher education institutions providing micro-credentials and digital badges continues to grow. As such, the purpose of this paper is to address the following research questions: (1) What is the prevalence of micro-credentials offered by universities?, (2) What are the characteristics of universities offering micro-credentials generally and information systems (IS) micro-credentials specifically?, and (3) What is the content of IS credentials?

2. RELATED LITERATURE

While information systems (IS) education has consistently been concerned with the alignment between IS curriculum and traditional industry-based certifications from well-known organizations such as Microsoft, CISCO, and Oracle (e.g., Gomillion, 2017; Marquardson & Elnoshokaty, 2020; Reinicke & Janicki, 2013; White, 2006), a new trend emerged beginning around 2012 centered on the concept of digital badges (Abramovich, Schunn, & Higashi, 2013; Ahn, Pellicone, & Butler, 2014; Beattie, 2014; Carey, 2012) and later related to the idea of micro-credentials (Damast, 2016; Rubleke & Cata, 2017) continuing into current day (Ermicioi, Liu, & Murphy, 2021; McGovern & Gogan, 2021; Pike, Brown, West, & Zentner, 2020). Gibson, Ostashewski, Flintoff, Grant, and Knight (2015) define a digital badge as "a representation of an accomplishment, interest or affiliation that is visual, available online, and contains metadata including links that help explain the context, meaning, process and result of an activity" (p. 404). While the terms digital badge and micro-credential are often used interchangeably in the literature, Fanfarelli and McDaniel (2019) suggest subtle differences, in that a micro-credential "implies an additional level of validation by a central authority such as an academic institution, professional organization, governing body, or a similar entity with a reputation and a body of expertise to qualify them to do this validation" (p. 11). Casilli and Hickey (2016) emphasize that a micro-credential entails the completion of a sequence or small cluster of courses related to a specific area of expertise. Micro-credentials may be obtained from any number of non-technical (e.g., project management, play therapy, entrepreneurship, etc.) or technical (e.g., cybersecurity, Excel, SQL, data analytics, etc.) areas.

While it was certainly true early on that micro-credentials were primarily the domain of companies such as Coursera, Udacity, and Udemy as opposed to universities (Gallagher, 2016), this is no longer the case as shown by the results of this paper. As universities (e.g., University of North Texas, University of Colorado Boulder, University of Illinois Urbana-Champaign) may be partnering with companies such as Coursera and others, many other universities are offering their own badges and micro-credentials directly to their students. The following literature serves as a representative sample of the ways micro-credentials and digital badges are being implemented in IS education.

Rigole and Hollingsworth (2017) conducted a study on the effect of badges on achievement and engagement in an information technology course in an eLearning environment. Two courses were included in the study, one fully online and the other offered in a hybrid format. The students were afforded the opportunity to earn digital badges by completing selected topics in digital media fundamentals. The same instructor taught each section and the same learning outcomes, curriculum, and schedule were used. Students earned badges for project-based assignments rather than for merely completing quizzes and discussion postings. In the course, students were able to earn up to six badges each representing one point toward their final course grade. Students were not required to complete the badges and participation was voluntary. However, additional badges were implemented in the course if students scored an 80% or better on the project-based assignments. The final course grade was used to measure engagement and achievement. The results indicated that 65% of students did not increase their final course grade due to completing the optional badges, while 35% did. In sum, the study showed an increase in engagement due to the implementation of badges, but not an increase in achievement by the majority of students.

To meet the needs of the constantly changing IS workplace, Rubleske and Cata (2017) argue that the implementation of an IS micro-credential program at the graduate-level is well suited. The purpose of which is to ensure that graduate IS programs become more focused on a specific area, remain up-to-date, and provide "a credential as an easily shareable and informationally transparent digital object (Rubleske & Cata, 2017, p. 1). In sum, IS graduate programs must become more "agile" in their approach to curriculum development in order to keep up with the rapid change of digital technology such that IS professionals need to gain new skills quickly and IS managers need to have a means by which to quickly and efficiently identify individuals with those skills. As such, Rubleske and Cata (2017) suggest that graduate IS programs "(1) teach the current IS management skills that employers need and (2) certify and describe in detail the credentials behind these skills" (p. 2). To support this notion, four characteristics of university micro-credentials are described and a set of value propositions are identified related to the various stakeholders including IS students, IS departments, and employers.

Cybersecurity is currently one of the fastest growing and popular areas of computing. There is a significant need, therefore, for well-prepared IS graduates. As such, Pike, Brown, West, and Zentner (2020) describe the development of a digital badging and e-portfolio environment in the area of cybersecurity education. The authors suggest that 50% of a student's learning should occur external to the requirements of their degree program through competency-based education, competitions, research, internships, and student clubs in conjunction with support and resources provided by faculty and the university. In an effort to provide such experiences, a student data center (SDC) and security operations center (SOC) were established. Both of these centers provide students with hands-on experience in a real-world environment. To track these activities, the use of digital badges is being explored. The idea is that students may earn badges for completion and assessment of various exercises which will then comprise an e-portfolio.

The basic idea behind the awarding of micro-credentials is to "stack" a series of certificates or courses in a related area. Ermicioi, Liu, and Murphy (2021) demonstrate how a digital badging system can be used to validate stackable certificates for micro-credentials in a graduate-level program. In a similar vein as Rubleske and Cata (2017), the authors apply the concept of micro-credentials to a graduate program in information technology (IT). They argue that "a more practical and sustainable approach" is needed (Ermicioi, Liu, & Murphy, 2021, p. 1). Using a conceptual framework of six forces previously developed by two of the authors (Liu & Murphy, 2012), a case study is developed describing the implementation of a digital badging program at the graduate level. By separating existing concentrations into certificates (i.e., cybersecurity, data science, digital health, digital transformation, project management), students are now able to complete the certificates individually and "stack" them to obtain a graduate-level certificate in a relatively short amount of time, for example, in two semesters. This provides students with a micro-credential before completing their graduate degree and potentially providing a beneficial credential in the event the student does not go on to complete the graduate degree. In essence, there is value-added regardless of completion. The authors conclude, "Given the need for reskilling and upskilling in the fast-moving IT field, digital badging of micro-credentials in the academic community is a must" (Ermicioi, Liu, & Murphy, 2021, p. 8).

The growth of digital badges and micro-credential is nowhere more evident than in the area of computing and technology. With growth in the popularity of data analytics, data science, and cybersecurity, for example, well-known technology companies such as Google have joined the ranks of private companies and universities as those who are now offering badges and micro-credentials (Katz, 2021). Consequently, IS education is fertile ground for the exploration of digital badging and micro-credentials. Moreover, IS education may be faced with making a decision about whether to compete or collaborate with the micro-credential and digital badge movement. As such, we believe the questions examined and the ensuing discussion and conclusions within this paper potentially make an important contribution to IS education.

3. METHODOLOGY

First, to establish a convenience sample for this study, data pertaining to all 1,860 universities listed on US News & World Report as of March 2, 2022, was scraped. US News & World Report was selected because it is arguably the most comprehensive list of US-based universities, despite questions surrounding its ranking methodology (Hartocollis 2022). The scraped data included the name of each university along with its ranking, funding type (i.e. private non-profit; private for-profit; public), setting (i.e. city, urban, suburban, rural, and unspecified), age, and median starting salary of graduates. While US News & World has one overall ranking, it assigns actual rank numbers based on an institution's classification (e.g. National Universities, National Liberal Arts Colleges, Regional Universities Midwest/North/South/West, Regional Colleges Midwest/North/South/West). Instead of trying to infer rank numbers, we simply grouped the institutions into three tiers based on US News & World's own classification: Tier 1 (national) includes all institutions classified as National Universities or National Liberal Arts Colleges. Tier 2 (regional) includes all institutions classified as Regional Universities Midwest/North/South/West or Regional Colleges Midwest/North/South/West. Lastly, tier 3 (unranked) includes all institutions not classified as either tier 1 or tier 2.

Second, the university names obtained in the previous step were used to scrape the Credly profile page for each university, if it existed. This was possible due to the fact that Credly uses a standardized URL schema in the form of https://www.credly.com/organizations/$university_name/badges whereby $university_name is replaced with the URL encoded name of a university (e.g. grand-valley-state-university for Grand Valley State University). The data scraped from each Credly profile page, if existed, included the number of badges offered as well as the links to each badge offered. In turn, the profile page of each badge offered was scraped to capture the description of the badge and the skills obtained.

Third, we borrowed from the IS competency realms identified in IS2020 A Competency Model for Undergraduate Programs in Information Systems, in order to ground the technology-related micro-credentials within the context of IS education in an effort to increase the relevancy of our study for IS educators. IS2020 represents the latest attempt by the Association of Information Systems (AIS) and Association for Computing Machinery (ACM) to put forth a model IS model curriculum. IS2020 differs from previous IS model curriculum recommendations (e.g., IS2010) by focusing on required competency areas rather than a set of traditional learning objectives. The IS competency realms include: Foundations, Data and Information Management, Technology and Security, Development, Organizational Domain, and Integration (Leidig & Salmela, 2020).

Finally, using content analysis (Berg, 2001) we analyzed the Credly data pertaining to badges in order to separate technology related micro-credentials from non-technology related credentials and to classify technology related credentials according to the IS competency realms. The first step in the analysis process consisted of a review of the micro-credential name. Where there was disagreement between the authors, the micro-credential descriptions were analyzed, and further discussion ensued when necessary. In sum, the authors, through an iterative process, were able to come to agreement on the separation of technology related and non-technology related micro-credentials. The second step in the analysis was to categorize the technology-related micro-credentials according to the IS competency realms. To facilitate this process, the authors first individually categorized the technology related micro-credentials according to the IS competency realms independently of one another. When disagreements arose, the authors discussed in detail to come to a consensus on the appropriate categorization. The findings provided in the Results section reflect the final categorizations. Lastly, the skills obtained from Credly were combined using simple word matching. So, for example, cybersecurity awareness, cybersecurity controls, cybersecurity management, cybersecurity policies were all combined into cybersecurity. Likewise Excel formulas and functions, Excel Pivot Tables, Excel spreadsheets, and Excel VBA were all combined into Excel. This allowed for a slightly higher-order analysis of skills.

4. RESULTS

Prevalence of Micro-Credentials

Of the 1,860 universities listed on US News & World Report, 127 (6.8%) offer micro-credentials on Credly. Of the 127 universities offering micro-credentials on Credly, 53 (41.7%) offer IS micro-credentials. On Credly, 127 universities offer a total of 2,308 micro-credentials. A relatively small number of universities offer a large number of micro-credentials. In fact, of the 127 universities offering a micro-credential on Credly, 114 (89.8%) offer more than one micro-credential. On average, a university on Credly offers 53 micro-credentials (M = 53.31, SD = 43.74). The most prolific universities on Credly are Grand Valley State University (n = 184), followed by University of North Dakota (n = 165), and Northeastern University (n = 150).

Characteristics of Universities Offering Micro-Credentials

Table 1 provides a cross-tabulation of three university tiers (i.e. tier 1 – national; tier 2 – regional; tier 3 – unranked) and the three groups of universities (i.e. all universities listed on US News & World Report; universities listed on US News & World Report that offer micro-credentials on Credly; universities listed on US News & World Report that offer IS micro-credentials on Credly). If universities offering micro-credentials or IS micro-credentials were no different from all universities with regards to the tiers they belong to, then one would expect a similar distribution of frequencies in each of the three columns.

	Number of All Universities	**Number of Universities Offering Micro-Credentials**	**Number of Universities Offering IS Micro-Credentials**
	n (%)	*n (%)*	*n (%)*
Tier 1 – National	626 (34%)	67 (53%)	29 (55%)
Tier 2 – Regional	840 (45%)	47 (37%)	19 (36%)
Tier 3 – Unranked	394 (21%)	13 (10%)	5 (9%)
Total	**1,860 (100%)**	**127 (100%)**	**53 (100%)**

Table 1: Comparison of University Tiers

A chi-square test of independence showed that there is a significant association between the three university tiers (i.e. tier 1; tier 2; tier 3) and the three groups of universities (i.e. all universities; universities offering micro-credentials; universities offering IS micro-credentials) ($X^2(4)$ = 30.84, p < .001). Thus, compared to all universities, universities offering micro-credentials or IS micro-credentials are more likely to belong to tier 1 – national.

Table 2 provides a cross-tabulation of three university funding types (i.e. private, non profit; private, for-profit; public) and the three groups of universities (i.e. all universities listed on US News & World Report; universities listed on US News & World Report that offer micro-credentials on Credly; universities listed on US News & World Report that offer IS micro-credentials on Credly). If universities offering micro-credentials or IS micro-credentials were no different from all universities with regards to their funding type, then one would expect a similar distribution of frequencies in each of the three columns.

	Number of All Universities	**Number of Universities Offering Micro-Credentials**	**Number of Universities Offering IS Micro-Credentials**
	n (%)	*n (%)*	*n (%)*
Private, Non-Profit	1094 (59%)	59 (46%)	22 (42%)
Private, For-Profit	58 (3%)	0	0
Public	708 (38%)	68 (54%)	31 (58%)
Total	**1,860 (100%)**	**127 (100%)**	**53 (100%)**

Table 2: Comparison of University Funding Types

A chi-square test of independence showed that there is a significant association between the three university funding types (i.e. private non-profit; private for-profit; public) and the three groups of universities (i.e. all universities; universities offering micro-credentials; universities offering IS micro-credentials) ($X^2(4)$ = 23.34, p < .001). Thus, compared to all universities, universities offering micro-credentials or IS micro-credentials are more likely to be public.

Table 3 provides a cross-tabulation of five university settings (i.e. city setting; urban setting; suburban setting; rural setting; unspecified setting) and the three groups of universities (i.e. all universities listed on US News & World Report; universities listed on US News & World Report that offer micro-credentials on

Credly; universities listed on US News & World Report that offer IS micro-credentials on Credly). If universities offering micro-credentials or IS micro-credentials were no different from all universities with regards to their setting, then one would expect a similar distribution of frequencies in each of the three columns.

	Number of All Universities	Number of Universities Offering Micro-Credentials	Number of Universities Offering IS Micro-Credentials
	n (%)	*n (%)*	*n (%)*
City Setting	334 (18%)	26 (20%)	8 (15%)
Urban Setting	342 (18%)	31 (24%)	16 (30%)
Suburban Setting	491 (26%)	44 (35%)	20 (38%)
Rural Setting	360 (19%)	16 (13%)	5 (9%)
Unspecified Setting	333 (18%)	10 (8%)	4 (8%)
Total	**1,860 (100%)**	**127 (100%)**	**53 (100%)**

Table 3: Comparison of University Setting

A chi-square test of independence showed that there is a significant association between the five university settings (i.e. city, urban, suburban, rural, and unspecified) and the three groups of universities (i.e. all universities; universities offering micro-credentials; universities offering IS micro-credentials) ($X^2(8) = 26.94$, $p < .001$). Thus, compared to all universities, universities offering a micro-credential or an IS micro-credential are more likely to be located in a city, urban, or suburban setting.

Finally, table 4 shows the average age of the university and average starting salary of graduates for each of the three groups of universities (i.e. all universities listed on US News & World Report; universities listed on US News & World Report that offer micro-credentials on Credly; universities listed on US News & World Report that offer IS micro-credentials on Credly). If universities offering micro-credentials or IS micro-credentials were no different from all universities with regards to their age or starting salary, then one would expect similar averages in each of the three columns.

	All Universities	Universities Offering Micro-Credentials	Universities Offering IS Micro-Credentials
	m (SD)	*m (SD)*	*m (SD)*
Age of the University	123.20 (48.69)	125.43 (42.09)	121.32 (44.49)
Starting Salary of Graduates	$47,342.35 ($10,121.33)	$50,095.04 ($7,120.88)	$51,008.00 ($7,689.67)

Table 4: Comparison of Age of the University and Starting Salary of Graduates

A one-way ANOVA showed that the difference in age of the university between the three groups of universities (i.e. all universities; universities offering micro-credentials; universities offering IS micro-credentials) is not significant ($F(2) = 0.17$, $p = .84$). Thus, compared to all universities, the age of universities that offer micro-credentials or IS micro-credentials tends to be the same.

A one-way ANOVA showed that the difference in starting salary of graduates between the three groups of universities (i.e. all universities; universities offering micro-credentials; universities offering IS micro-credentials) is significant ($F(2) = 7.79$, $p < .001$). A Tukey post-hoc test revealed that the differences in starting salary are significant between all universities and universities offering a micro-credential ($\Delta = \$2,752.69$, $p < .05$) as well as between all universities and universities offering an IS micro-credential ($\Delta = \$3,665.65$, $p < .05$). However, the difference in starting salary between universities offering a micro-credential and universities offering an IS micro-credential is not significant ($\Delta = \$912.95$, $p = .84$). Thus, compared to all universities, the starting salary of graduates at universities that offer a micro-credential or an IS micro-credential tends to be higher.

Content of Micro-credentials

Of the 2,308 micro-credentials offered by universities on Credly, 195 (8.4%) can be considered IS micro-credentials as they address competency realms defined in the IS2020 curriculum. The top three IS2020 competency realms addressed by IS micro-credentials are technology, data, and development. The distribution of IS micro-credentials for each IS2020 competency realm is shown in figure 1 below.

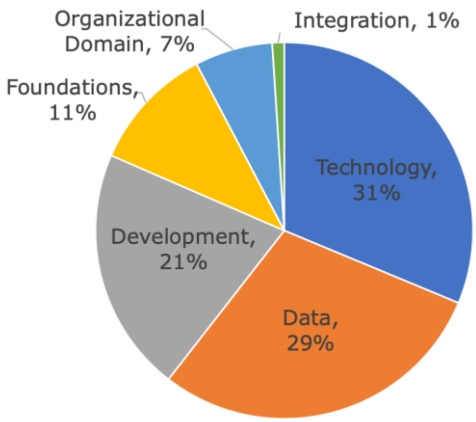


Figure 1: Classification of IS Micro-credentials by IS2020 Competency Realm

Lastly, each micro-credential on Credly has one or more skills associated with it. The most frequently associated skills among IS micro-credentials are cybersecurity (7.4%), Excel (3.9%), AI/ML (3.2%), SQL (2.7%), and HTML, CSS, JS (2.6%).

5. DISCUSSION AND CONCLUSION

Based on the results of this study, it is apparent that only very few universities are currently offering micro-credentials. The few universities that do offer micro-credentials tend to offer numerous micro-credentials. Moreover, the few universities that do offer micro-credentials tend to be highly-regarded and widely-known. These universities can be considered early adopters in diffusion of innovations theory (Rogers, 2003), as they are likely to hold strong positions of opinion leadership among their peers. Interestingly, only a small subset of micro-credentials focuses on IS content as defined in the IS2020 curriculum. Clearly, universities have yet to widely adopt micro-credentials in general and IS micro-credentials in particular. Given the wide adoption of micro-credentials in the private sector, it is likely that the majority of universities will soon follow suit.

Contributions

This study makes several important practical and theoretical contributions. First and foremost, it provides current and in-depth empirical insights into the prevalence, characteristics, and content of micro-credentials in general and IS micro-credentials specifically. These insights can be leveraged by organizations in the private sector, such as digital credentialing platforms and training providers, as well as by universities to develop new or adjust current offerings in the micro-credentialing space. From a theoretical perspective, insights gleaned into the adoption of micro-credentials, which is an innovation in higher education, by certain types of universities, might add to the diffusion of innovations theory (Rogers, 2003).

Limitations

This study is not without shortcomings. First, this study was based on a snapshot of data collected in March 2022. It is possible that universities launched or discontinued micro-credentials shortly before or after the data collection, which wouldn't be captured in this study. Moreover, this study doesn't include longitudinal data and as a result can't analyze the development of micro-credential offerings over time. Second, this study used all 1,860 universities listed on US News & World Report (n.d.) as a convenience sample. While this probably provides a comprehensive sample of US-based universities, it excludes all non-US-based universities. Third, this study cross-referenced all 1,860 universities listed on US News & World Report with the names of organizations that issue digital credentials on Credly (n.d.). Although Credly is one of the world's largest digital credentialing platforms, having issued more than 50 million credentials to 25 million people worldwide, it's certainly not the only one. Hence it's possible that universities in this study's convenience sample use other or additional digital credentialing platforms which wouldn't be captured in this study. Finally, this study used content analysis to categorize micro-credentials into IS2020 competency realms. Though great care has been taken to ensure a repeatable and objective process, it's possible that the authors shared preconceived notions or misunderstandings that led to misclassifications.

Future Research

To address some of this study's limitations, future research may wish to collect longitudinal data. Already capturing just a second dataset and comparing it with this study would allow drawing certain conclusions regarding the development of micro-credentials being offered by universities over time. Moreover, future research should include international universities in order to gain insights regarding differences between geographic regions. Furthermore, future research should collect data from other digital credentialing platforms in order to potentially capture more micro-credentials. Lastly, future research might benefit from using automated data collection and analysis methods that would allow for faster and potentially more objective results.

6. REFERENCES

Abramovich, S., Schunn, C., & Higashi, R. M. (2013). Are Badges useful in Education?: It Depends Upon the Type of Badge and the Expertise of the Learner. *Educational Technology Research and Development*, 61(2), 217-232.

Ahn, J., Pellicone, A., & Butler, B. (2014). Open Badges for Education: What are the Implications at the Intersection of Open Systems and Badging? *Research in Learning Technology*, 22, 1-12.

Ayala, C., Dick, G., & Treadway, J. (2014). The MOOCs are Coming! Revolution or Fad in the Business School?. *Communications of the Association for Information Systems*, 35(12), 225-243.

Beattie, S. (2014, June 14). Types of Digital Learning Badges: Drawing on the Xbox Achievement Experience [Web blog post]. Retrieved from HASTAC.org website at https://www.hastac.org/blogs/scott-beattie/2014/06/30/types-digital-learning-badges-drawing-xbox-achievement-experience

Berg, B. L. (2001). *Qualitative Research Methods for the Social Sciences* (4th. ed.). Boston, MA: Allyn and Bacon.

Carey, K. (2012, April 8). A Future Full of Badges. *The Chronicle of Higher Education*. Retrieved from https://www.chronicle.com/article/a-future-full-of-badges/?bc_nonce=ypn1wukqefovop2fup1qks&cid=reg_wall_signup

Casilli, C., & Hickey, D. 2016. Transcending Conventional Credentialing and Assessment Paradigms with Information-Rich Badges. *The Information Society*, 12(2), 117-129.

Casilli, C., & Knight, E. (2012). 7 Things You Should Know About Badges. EDUCAUSE. Retrieved from http://www.educause.edu/library/resources/7-things-you-should-know-about-badges

Credly (n.d.). Digital Credentials. Retrieved March 2, 2022 from https://www.credly.com/

Crossley, M. (2021, February 7). Micro Credentials Not the Education Revolution They Were First Touted to Be. The Sydney Morning Herald. Retrieved from https://www.smh.com.au/business/workplace/micro-credentials-not-the-education-revolution-they-were-first-touted-to-be-20210223-p57558.html

Damast, A. (2016). The micro-credential comes of age. Poets & Quants for Executives. Retrieved from https://poetsandquantsforexecs.com/news/5533/

Ermicioi, N., Liu, M., & Murphy, D. (2021). Digital Badges as an Agile Pathway: Implementing Graduate-Level, Micro-Credential Programs to Reskill the IT Workforce. *Proceedings of the 2021 EDSIG Conference*, 7(5507), 1-11.

Fanfarelli, J., & McDaniel, R. (2019). Designing Effective Digital Badges: Applications for Learning (1st ed.). Routledge. Retrieved from https://www.researchgate.net/publication/331903962_Designing_Effective_Digital_Badges_Applications_for_Learning

Gallagher, S.R. (2016). The Future of University Credentials: New Developments at the Intersection of Higher Education and Hiring. Harvard Education Press, Cambridge, MA.

Gibson, D., Ostashewski, N., Flintoff, K., Grant, S., & Knight, E. (2015). Digital Badges in Education. *Education and Information Technologies*, 20(2), 403–410.

Gomillion, D. L. (2017). The Role of Industry Certifications in an AACSB-Accredited Institution. *Information Systems Education Journal*, 15(1), 68-79.

Hartocollis, A. (2022). Columbia Loses Its No. 2 Spot in the U.S. News Rankings. *New York Times*. Retrieved July 9, 2022 from https://www.nytimes.com/2022/07/08/us/us-news-rankings-columbia.html

Holton, K. (2022). EXCLUSIVE: Pearson buys certification group Credly in deal valued at $200m. Retrieved July 1, 2022 from https://www.reuters.com/business/exclusive-pearson-buys-certification-group-credly-deal-valued-200m-2022-01-31/

Katz, J. L. (2021, May 3). What You Need to Know About Google Career Certificates. *U.S. News & World Report*. Retrieved from https://www.usnews.com/education/google-career-certifications

Leidig, P., & Salmela, H. (2020). IS 2020 A Competency Model for Undergraduate Programs in Information Systems. 5-184. Retrieved from https://is2020.hosting2.acm.org/2021/02/07/is2020-final-report/

Liu, X., & Murphy, D. (2012). Tackling an IS Educator's Dilemma: A Holistic Model for "When" And "How" to Incorporate New Technology Courses into the IS/IT Curriculum. *Proceedings of the Southern Association for Information Systems Conference*, 31, 176-181.

Marquardson, J., & Elnoshokaty, A. (2020). Skills, Certifications, or Degrees: What Companies Demand for Entry-Level Cybersecurity Jobs. *Information Systems Education Journal*, 18(1), 22-28.

McGovern, T. M., & Gogan, J. L. (2021). Can Digital Badging Help Universities Flexibly Support Students and Faculty During Crises? A Proposal and Call for Research. *Proceedings of the 54th Hawaii International Conference on System Sciences*, 6339-6348.

Pike, R. E., Brown, B., West, T., & Zentner, A. (2020). Digital Badges and E-Portfolios in Cybersecurity Education. *Information Systems Journal*, 18(5), 16-25.

Reinicke, B. A., & Janicki, T. (2013). Who Needs Certifications? A Survey of Certifications in the IT Industry. *Proceedings of the Information Systems Educators Conference*, 30(2539), 1-8.

Rigole, N., Hollingsworth, L., & Ray, J. (2017). Badges and Gamification In eLearning: Effects on Achievement And Engagement. *Proceedings of the Southern Association of Information Systems*, 13, 1-5.

Rogers, E. M. (2003). Diffusion of Innovations, 5th Edition. Florence, MA: Free Press.

Rubleske, J., & Cata, T. (2017). University Micro-Credentials and the Need for Agile IS Skill Development Programs. *Proceedings of the EDSIG Conference*, 3(4379), 1-9.

THECB (2022). Accelerating Credentials of Purpose and Value Grant Program. Retrieved from https://www.highered.texas.gov/institutional-resources-programs/institutional-grant-opportunities/accelerating-credentials-of-purpose-and-value-grant-program/

US News & World Report (n.d.). Search US News Best Colleges. Retrieved March 2, 2022 from https://usnews.com/best-colleges/search?_mode=table

White, G. L. (2006). Vendor/Industry Certifications and a College Degree: A Proposed Concentration for Network Infrastructure. *Information Systems Education Journal*, 4(48), 3-7.

User Experience Design in The Information Systems Curriculum: Lessons Learned and Best Practices

Karoly Bozan
bozank@duq.edu

Claire Stoner
Stonerc@duq.edu

Burcu Maden
madenb@duq.edu

Duquesne University
Pittsburgh, PA, USA

Abstract

User Experience Design (UXD) is an often-neglected area of the information systems (IS) curriculum. UXD classes specifically designed for IS students are still uncommon in IS programs and this study aims to add to the body of knowledge to prepare a more well-rounded future generation of IS professionals. With this goal in mind, this study describes the redesign of an introductory UXD course following Kolb's learning cycle and constructivist instructional models. This paper describes the implementation of the supporting pedagogy and the opportunity for students to better master core UXD concepts. UXD, a multi-disciplinary area is built on skills learned in systems analysis and design class while students learn to apply relevant concepts through a hands-on, instructor-led, individual, in-class sample project. The skills are then applied by engaging students in active learning in a team setting to deliver value to a local organization by solving related, real-life challenges. Students work on community-engaged team projects to enhance their appreciation of the impact and relevance of their semester-long project deliverable. Constructivism guides the instructional models of the framework, in which problem-based learning is used to help students build and apply relevant skills. The instructional models and implications for instructional design are discussed along with a proposed pedagogical approach, course setting and structure, tools and techniques engaged, student feedback analyzed, and lessons learned.

Keywords: User experience design, IS Curriculum, Project-based learning, Community-engaged learning, Constructivism

1. INTRODUCTION

The recent technological advancements contribute to the ability of software development organizations (SDOs) to meet the changing user expectations. Users expect more than the sheer ability to complete a task through a software application, they also expect the process to provide a meaningful, engaging, and relevant experience that ultimately results in greater user satisfaction with less cognitive effort (Dadarkar & Tiwari, 2022). These user expectations have been turning the attention of SDOs to user experience design as a dynamically morphing domain as part of their product design. User experience is the experience of technology, products or services that refers to something larger than usability or one of its dimensions such as satisfaction or attitude (McCarthy, 2004); it is the emotions that the user encounters while using a service, a product or an application (Dirin, 2018).

SDOs view UXD from two angles. First, from the customer's point of view UXD provides an opportunity for competitive advantage, improved return on investment. Customers expect products that fulfill their needs to accomplish a task in an effective and efficient manner. Positive customer experience is found to increase customer loyalty. (Cheng, 2020). Second, from the internal operation's point of view UXD has the promise to provide increased business process efficiency, improved employee productivity, lower training costs, reduced error rates, and more secure systems to support internal operations (Larson & Harrington, 2012).

The demand for UXD professionals is further fueled by global organizations and startups that aim to build the next generation of websites, apps, and IoT devices. Therefore, a well-rounded IS professional should possess the basic understanding of UXD principles to properly address the design-based aspect of information systems. The response to this demand is lacking in most business schools as part of their undergraduate IS programs (MacDonald, Rose, & Putnam, 2022). We found from a random review of 120 large, medium, and small universities that approximately 25% of schools with IS curriculums have course with UXD focus. Please see Appendix A for a full list of reviewed schools. While it is always challenging to augment the already packed IS curriculum covering technical and business aspects of IS degree requirements, market demands for UXD professionals in the software industry should not be ignored (Getto & Beecher, 2016; Benyon, 2019; Lauer & Brumberger, 2016).

UXD is a multidisciplinary area that magnifies the effort to design a course that best serves IS students. The effort to design, develop, and implement UXD into the IS curriculum is further amplified given the variety of students' skills, backgrounds, and other IS courses across universities. Textbooks are virtually non-existent that cover the UXD for IS students and the related books mainly cover UXD as part of human-computer interaction (HCI) with deeper roots in cognitive psychology and ergonomics geared toward academic fields than practice (Gull, Saeed, & Iqbal et al., 2018). Hence, HCI is more theoretical while UXD is more practical with hands-on project opportunities for IS students. Domain specific UXD books include less relevant and outdated examples with very limited opportunities for the students to practice the concepts covered. In order to be successful in the IS profession, students need to learn relevant and up-to-date skills through engaging projects that allow students "learning by doing" (Smith, 2021).

In response to the steady growth of industry demand and to develop more well-rounded IS graduates, we designed a standalone UXD course that builds on prior IS courses and motivates students to explore the impact UXD has on users of information systems. Students will then use the UXD principles in their IS Capstone course where they develop an app in response to an organization challenge. Learning from the first semester experience, we methodologically re-designed the course with focus on active learning. With this pedagogical approach, students have the opportunity to explore UXD principles by connecting their prior knowledge to the new concepts, apply the concepts in a project setting and contribute in different roles in a self-organized team environment (Spielhofer & Haselberger, 2021). This overarching progression of experimental learning and applying skills was well received by students and our analyses found increased engagement and motivation. In the following sections we review relevant literature, the implementation of our pedagogical approach and the result of our analyses based on data from before and after the pedagogical change followed by lessons learned.

2. LITERATURE REVIEW

This literature review aims to compile UXD specific scholarship of teaching and learning (SoTL) yet occasionally we had to broaden our search and include HCI literature beside IS educational journals (Kreber, 2007; Witman, Ritchlin, & Arboleda, 2007).

UXD in IS
UXD is a complex, multi-disciplinary domain rooted in industrial design with human ergonomics and other cognitive psychological constituents (Churchill, Bowser, & Preece, 2013). Not surprisingly, IS programs scarcely incorporate UXD classes into their curriculums besides basic usability and user interface design principles (Getto, Potts & Salvo, 2013; Haaksma, de Jong & Karreman, 2018). User-centricity is a premise of digitalization and the recent IS literature is more directly pointing out the importance of producing software applications,

which not only considers the business goals, but also the client expectations (Kocielnik, Amershi & Bennett, 2019). Brenner, Osterle & Petrie et al. (2014) describe the digital user as a new design perspective that shifts the focus to the individual users and their needs throughout the software development life-cycle. Universities and professional bodies such as the Association for Computing Machinery (ACM) are challenged to keep their curricula up-to-date to reflect the changing sociocultural characteristics of users. Getto & Beecher (2016) and Altay (2014) have noted that IS programs are still challenged to understand the specific UXD competencies and they call for educational programs at universities, which include working knowledge of UXD principles and processes of digital products and services by solving real organizational challenges. The ACM Curricula Report of 2020 and the IS 2020 A Competency Model for Undergraduate Programs in Information Systems have recognized UXD as part of all four competency subgroups and described five distinct draft competencies (Computing Curricula, 2020; ACM/AIS Task Force, 2020).

UXD in Capstone
The IS Capstone course is designed for IS students during the end of their program of study to showcase their relevant skills (Abrahams, 2010). The class often includes a semester-long project on which students work in a team setting with instructor supervision and provides the students the opportunity to work on a solution in response to business challenges of practice (Payne, Flynn, & Whitfield, 2008). The capstone course with the purpose of empowering students to evaluate, appreciate, and apply multiple skills and perspectives in the form of a collaborative project is a common accreditation requirement among colleges (Knox & Nairn, 2021).

UXD in Team Setting
A common method of delivering UXD courses is the emphasis on teamwork. Dividing students into groups allows them to be more engaged and contribute by applying the techniques, methods, and skills they learned in the course complemented by their diverse backgrounds and experiences. Cliburn (2017) describes the team-based learning principle method of their course, in which students collaborated in the design, development, and evaluation of interactive applications. In Cliburn's class students proposed a project and then they "applied" for different roles on the project based on their skills and strengths. Then they followed a methodology to understand client needs and develop an application in response to the identified needs. This approach suggests that developing a highly-effective, self-organizing team is modeled in UXD courses.

UXD in Community Engaged Learning
Community engaged learning (CEL) is gaining increasing importance in higher education and is a strategic initiative at the author's institution. This approach is also known as service learning, an educational strategy, which combines classroom learning with a relevant community service experience, especially in the IS Capstone course (Wei, Siow, & Burley, 2007; Preiser-Houy & Navarrete, 2012). Mulder (2015) describes their students' collaboration with local community stakeholders to solve societal problems, yet their work was related to HCI curriculum (Gull et al., 2018). The benefit of CEL has been discussed in the literature, e.g.(Rose, 2005) but it lacks UXD specific context. This provides an opportunity to harness CEL's impact on undergraduate IS students' learning experience. Coupled with CEL, UXD is a suitable subject matter that could be incorporated into the IS curriculum body of knowledge.

Learning Models that Support UXD
Constructivist learning theory is built on the premise of building on previously established knowledge (Hein, 1991). This principle allows students to build personal interpretation of the concepts taught in the classroom based on their experiences and interactions with others. The goal of constructivist instruction should support the active process of knowledge construction rather than communicating knowledge (Connolly & Begg, 2006). This can be manifested in engaging students in the actual use of the concepts and techniques in real business scenarios rather than structured learning for the task.

Several instructional models are appropriate to support active or experiential learning. First, a resource-based view of learning allows the instructor to assume a guiding role providing resources, a shift from dispensing expert knowledge (Rakes, 1996). Second, inquiry-based learning allows students to find a solution through research and asking questions driven by a goal initially provided by the instructor (Hu, Kuh, Li, 2008). Teaching UXD presents the opportunity for inquiry-based learning as students need to gather needs and identify areas for improvement through interaction with stakeholders, within the team and with the instructor.

To support the project portion of the UXD class, problem-based learning (PBL) engages students with the process that begins with a problem to be solved rather than content to be mastered (Khair, Skudai, & Malysia et al. 2011). PBL engages students through encouraging them to construct their own understanding of the situation, link it to their experience and prior knowledge while interacting with others to refine their understanding and eventually solve the problem (McCarthy, Grabowski, & Koszalka, 1998). PBL helps students develop reasoning skill, self-directed learning strategies, and collaborative learning skills (Khair et al. 2011). Most constructivist teaching strategies place heavy emphasis on collaborative or cooperative learning (CL), which demonstrates the notion that a solution to a problem is not a function of an individual but rather of the distributed intelligence of diverse contributors. (Roblyer, Edwards, & Havriluk, 1996). Driscoll (2000) points out that CL helps students to appreciate varying views even those outside of their own.

Teaching UXD is well positioned for the above learning models yet there is a gap in the relevant scholarship of teaching and learning literature to guide IS instructors on the design and implementation of UXD into their programs.

Challenges with Implementing Innovative UXD Teaching Models

The early and more recent relevant literature point out the dynamic nature of the discipline, related tools and technology. These changes require continuous effort to use updated textbooks, create practical assignments and utilize the latest tools (Sousa Santos, 2006; Talone, Basavaraj, & Wisniewski, 2017). To overcome these challenges, it is recommended that the methods, tools, and techniques used in class are harmonized with industry demands and assignments reflect the current needs and expectations of users. Shumba (2006) pointed out the challenge with assigning students to teams with the right composition of complementary skillsets. Students often possess similar skills from prior classes and the narrower breadth of responses to a problem often lacks creativity. Furthermore, collecting requirements and communicating with external stakeholders is often a challenge for CEL projects (Comeau et al., 2019; Brubaker et al., 2018).

Our literature review of SoTL in the IS journals listed in Witman et al. (2007) resulted in very limited results that covered UXD and even HCI components of IS programs. Hence our motivation to re-design our UXD course and report our methodology, pedagogical approach and lessons learned.

3. PEDAGOGICAL APPROACH

In response to the gaps in the existing UXD education literature and feedback from students, we identified opportunities to re-design the UXD element of our IS curriculum. The areas covered for the re-design are: UXD in IS, UXD in Capstone, collaboration in teams, UXD in CEL, and learning models. Improvements are discussed in the next sub-sections and summarized in Table 2.

Competency	UXD Cycle Phase	Specific Coverage
Design tools and techniques	UCD Cycle established	Four distinct phases with unique techniques and tools.
Stakeholder needs	Requirements gathering (business and user needs)	Observation, interview, focus group, survey, etc. and document findings
Benchmark and Standards	Alternative design	ScOUt model, Task-orientation, UX Honeycomb
Integrative Design	Prototyping	Navigation and Information Architecture
Application Design	Evaluation	Techniques to capture qualitative and quantitative feedback for analysis and improvement

Table 1: ACM Recommended Competencies Embedded in the UXD Course

UXD in IS

First, we followed the ACM recommended competencies (Computing Curricula, 2020, pg. 119) namely, design tools and techniques, stakeholder needs, benchmarks and standards, integrative design, and application design. Table 1. summarizes the competencies recommended in the ACM Computing Curricula and the areas of class covering them.

We employed the UXD Cycle, which includes four distinct phases, namely: requirement gathering, alternative design, prototyping, and evaluation. The requirement gathering established the problem space where the user needs and challenges are identified. The alternative design phase helped students to identify the design space, the specific area of problem(s) that they aim to solve with an enhanced design. In the prototyping phase students propose enhanced ways for users to accomplish their task. In the evaluation phase, users have the opportunity to test the redesigned process of accomplishing their task and providing feedback. These phases corresponded with the competency guidelines detailed in the ACM Curricula for UXD in the four disciplines (computer engineering, computer science, information systems, and software engineering). Our review of current course description for other schools with UXD course(s) found that only a portion followed this methodological approach. The techniques and phases will be discussed in the Implementation of Pedagogy section.

Instructional Model

We followed Kolb's (1984) learning style quadrants and experimental learning cycle as a framework in our pedagogy design. Kolb's learning theory is built on the premise that learning new concepts is provided by new experiences: "Learning is the process whereby knowledge is created through the transformation of experience" (Kolb, 1984, pg. 38). Figure 1. summarizes the phases in Kolb's experiential learning cycle (ELC). Kolb's four stages are embedded in our UXD

UXD in CEL

In response to the author's University's Strategic Imperative of engaging with the community, we partnered with local organizations to provide UXD related services. Students worked on enhancing their users' experience in the form of re-designing their website (UXD course) and building an app to support internal operations (IS Capstone course). Students appreciated the involvement and the impact of their engagement with the community. This approach motivated students to feel ownership of their deliverable as they presented their proposal to the client on-site.

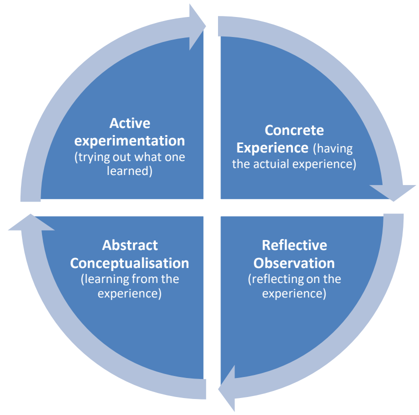


Figure 1. Kolb's (1984) Experiential Learning Cycle

In order to prepare students to solve the real-life business challenges they face, concepts are covered and re-iterated through an individual in-class mock project. The purpose for the individual assignments is to replicate scenarios in which the concepts can be used and students are prepared to apply their newly gained knowledge in team assignments. Therefore, they are presented with the opportunity to apply the knowledge twice and potentially build a deeper understanding of the subject matter.

Collaboration in Teams

Another aspect of our pedagogical approach is the team environment. Students often note in class evaluations that some team members do not contribute or passive participants in the planning sessions wait to be told what to do. We considered this feedback and the challenges identified in the literature review and implemented the following plans to alleviate these challenges. First, during team formation, we ensured diversity in team composition. Students filled out a brief survey in which they rated their skills and interest in a variety of soft and technical skills. We further considered their second degree or minor, gender, and other demographic characteristics. We identified that self-organizing teams are difficult to implement so instructor intervention was necessary. We announced that each assignment will be led by a rotating team lead who is responsible for setting up and leading meetings, submitting the deliverable and providing a transparent summary of each team members' contribution on the top of the assignment in the form of a contribution log. Team leads also had the opportunity to provide private input to the instructor of the challenges that they were not meant to handle such as non-contributing or non-responding team member(s). Students were asked to decide among themselves who takes on what part of the agreed sub-tasks and how it will contribute toward the deliverable. Team members filled out self- and peer assessments during the midterms and after submitting the final deliverable.

Furthermore, we utilized the latest technologies common in today's team environments for technology-supported collaborative learning. These technologies included Atlassian products of Confluence and Jira for work assignment and administration while students utilized Slack for communication. Instructors had access to monitor progress, activities, and student contributions in these applications.

UXD in Capstone
Students in the Capstone course were expected to utilize the knowledge they gained in the UXD class and the emphasis was on the collaboration aspect of the CEL project. Details of the Capstone course is outside of the scope of this paper.

Table 2. summarizes the challenges we identified in the literature review and the actions we took during the UXD class design and development.

4. IMPLEMENTATON OF THE PEDAGOGY

The pedagogical approach was implemented across two sequential courses for undergraduate IS students. The first course titled "User Experience Design" introduced students to UXD principles and is a pre-requisite for the second class, the "IS Capstone", in which students are required to apply the skills in their final project deliverable that draws together all related concepts covered in previous major courses of the IS curriculum. Students take these two subsequent courses during their junior and/or senior year.

As previously mentioned, there was no textbook available to follow for UXD in IS context so the instructor developed and introduced two general models with elements that students were already familiar with from other IS courses:

(1) a base model that covers the core concepts of UXD, and
(2) the UXD Cycle as a process model

Area (issues prior to redesign)	**Gap in literature**	**Action**
UXD in IS (not guided by competencies)	No evidence of ACM Competencies in IS programs	Included the UXD Cycle to cover the recommended five competencies
UXD in Capstone (no UXD component)	No evidence in the literature of UXD in IS Capstone courses	Students were required to apply knowledge from UXD class to enhance CEL project deliverable.
Teams (no directions from instructor)	Composition, contribution	Diverse teams to promote creativity, contribution log, self- and peer assessments, team lead, self-organization
UXD in CEL (no UXD component)	Only HCI in CEL in the relevant literature	Students worked with local organizations to provide UXD related solutions
Instructional Models (lecture heavy, limited in-class exercises)	Limited directions in literature for UXD course design	Kolb's ELC (inquiry- and problem-based)

Table 2: Response to Gaps in UXD in IS Program Literature

Combining these two guiding models and implementing across two IS courses is an innovative pedagogical approach to teach UXD for IS students. It allows the instructor to combine concepts learned in systems analysis and design with a front-end focus rather than system level-holistic approach. Even though students were familiar with the elements of the models the particular methodology developed for UXD allowed the instructor to highlight the key differences between systems design and UXD. Students showed more interest in applying previously established knowledge as a premise of constructivism (Hein, 1991). The instructor encouraged students to build personal interpretations of the concepts in the UXD methodology that enables the active process of knowledge construction. The project setting further supported the constructivist learning model as students linked their knowledge and

experiences with fellow students to construct their own understanding of the methodology to support UXD.
The next section describes the two models in detail and how they enable students' knowledge construction.

Core Concepts
A model was drawn to familiarize students with the core concepts of UXD. We collectively named it the ScOUt model as shown in Figure 2.:

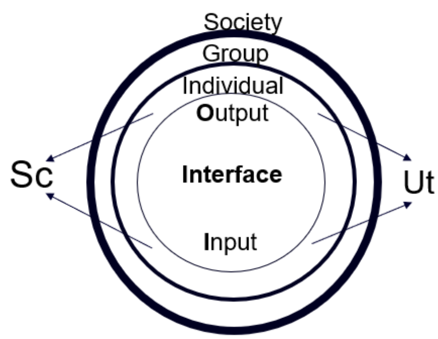


Figure 2. UXD Core Concepts (The ScOUt Model)

This model helps students conceptualize the notion that individuals that use some technology to accomplish a task need to interact with the technology and that experience needs to be enhanced. Users have a goal to accomplish a task (Ut) and they interact with the system's core functionality (Sc) through an Interface. Users must provide some input (I) that is easy to understand and leads to some output by the system (O) that completes the task based on Dix, Finlay, Abowd, & Beale (2004) and rooted in human-computer interaction but designed to cover IS concepts and examples. Group and societal expectations are considered in addition to individual needs. This basic model is revisited many times to ensure students do not lose sight of the process-based approach of UXD. Several examples from the industrial design are discussed with students, such as garage doors, chairs, and even Norman's Door and linked to software application design. For homework, students are asked to identify digital interfaces that fail to consider the above model, for example, difficulty to understand what input is required or possible (affordances and signifiers) or output is not providing the desired outcome or does not complete the task.

Students are then introduced to the UXD Cycle, which is a four phased process-based framework comprising of (1) requirement gathering or discovery; (2) Alternative Design; (3) Prototyping; and (4) Evaluation. Students spend several weeks on each phase of the UXD Cycle to learn specific tools, techniques, and methods for each phase and then apply in a mock in-class, individual project. A high-level overview of concepts and techniques covered for each phase in the UXD Cycle:
1, Requirements gathering: establishing the *problem space*. Students learn how needs are identified from both the business and users along with tasks that users want to accomplish in different scenarios. The appropriate use of different requirements gathering techniques are discussed (observations, focus groups, interviews, surveys). Then students learn to compile the information gathered for analysis and to present their findings. Students need to be able to describe who the users are through user characteristic tables, personas, etc. and how the users currently accomplish the tasks through flowchart, scenarios, essential use case analysis, hierarchical task analysis, current UI critique, etc.

The mock in-class project is based on data collected from students interviewing each other about their perceived college experience followed by instructor lead discussion to identify common themes across the interviews. Focus on good grades in order to get a good job is a theme that most students identified in their data and further focused brainstorming discussion identified potential solutions to help students get good grades in the form of an application that helps students better prepare for classes and tests by pairing them up with study groups. This example shows students that they may not know their needs until after several rounds of questions and answers, often performed by business or system analysts. The established goal is to build a "Study Buddy" app and further discussion establishes tasks students want to accomplish with this app. Related tools and techniques are covered that students use to present their findings about users and tasks that will serve the basis of identifying system features that help users accomplish those tasks in an efficient and effective manner. In this phase, students develop contextual understanding of the tasks and identify with related opportunities and constraints.

Clients from the community present their organization and challenges at this point of the course and students are equipped with the skills to ask relevant questions that help them identify who the users are, what their goals are and what tasks users want to accomplish. Again, they present their findings but now in teams and applying the techniques and tools covered through the in-class exercise.

2, Designing alternatives – establishing the *design space*: after establishing the requirements, students are able to identify the area they plan to improve through helping users to accomplish their intended task. The goal is to improve the way users currently achieve that task guided by sound requirements identified in the prior UXD cycle element. This is the step where improved user experience is established. Brainstorming and other relevant techniques and tools are used to capture ideas in a team setting after the basics are covered in-class.

3, Prototyping: In this phase students model various system features that meet core aspects of the tasks users want to accomplish. We differentiate low and high fidelity and horizontal and vertical prototypes. Low fidelity techniques include sketching, wireframing, storyboards, and card-based activities. Students enjoy this step as they can quickly pivot to different ideas without investing much effort or being bound to an idea. Hand-drawing is encouraged yet several applications are available that help to create low fidelity prototypes. The high-fidelity prototypes include close-to-final layout with interaction modeled. Users can "touch and feel" this prototype and provide more useful feedback.

4, Evaluation: We discuss formative and summative evaluation and assess learnability and memorability as quantitative measures while assessing cognitive and emotional measures as qualitative measures.

Students apply the skills in the real-life project and reach out to the client for early feedback to ensure they understand the needs as they progress in the design cycle. The iterative nature of the UXD cycle is emphasized in class as students often receive feedback that requires further improvement. This is an important lesson in the class that we have been emphasizing from the beginning as the first prototype will likely not be perfect.

Table 3. Summarizes the techniques used and artifacts create din class across the four phases of User Experience Design Cycle: (1) Requirements gathering; (2) Designing alternatives, (3) Prototyping; (4) Evaluation.

Assessments
Individual assignments include the steps to develop the "Study Buddy" app. The team assignments similarly include deliverables for each UXD Cycle phase with write up on methods, tools, techniques and concepts used, lessons learned, and improvements made. Furthermore, the quizzes tested students' understanding of the concepts and essay questions provided the opportunity for students to elaborate on scenario-based questions.

UXD Phase	Techniques	Artifact
1.	Naturalistic Observation Survey Focus Group Interview	Empathy and Affinity Map, User characteristic tables, Persona, Flowchart, Scenarios, Essential use case analysis, Hierarchical task analysis, Current UI critique
2.	Brainstorming, Mind mapping, SWOT, SCAMPER	Categorize, Reduce, Analyze: Choose Design Space, Functional and Non-functional requirements
3.	Low fidelity: sketching, storyboard, card-based, wireframe mid- and high-fidelity: software based, horizontal / vertical prototypes	Navigation design, information architecture, interaction design, conceptual model, wireframes, interface design
4.	Formative and Summative evaluation, questionnaires, log data analysis, interview	Analysis results on learnability, memorability, cognitive and emotional measures

Table 3: Techniques Learned and Artifacts Created in the UXD Class

The class included several related videos in which SDO executives explained the importance of UXD and videos in which frustrated users pointed out the poor UXD on everyday things. We also had two expert guest-speakers one from the industrial design field while the other is from a SDO.

5. ANALYSIS

At the end of the UXD and IS Capstone courses, students completed an attitudinal survey. Data from these surveys were used to compare students' perceived knowledge of a variety of IS elements in UXD context prior and after the class re-design. The redesigned Capstone course had an enrollment of 40 student while 64 students took the redesigned UXD course across two sections.

We present the findings from the UXD class as the redesign had a greater impact on this class. Following Baham (2019) we invited all students to complete an anonymous survey that measured their perceived UXD skills prior the class, current knowledge (after the class) of specific areas of the UXD cycle, and their comfort level with UXD moving forward. The measures were answered using a 5-point Likert scale ranging from 1-Strongly Disagree to 5-Strongly Agree and 3-Neutral. A total of 32 out of 35 students completed the survey prior to re-design while 55 out of 64 students in the re-designed course completed it. Data was collected at the end of the semester upon completion of the project. The full questions are in Appendix C. The results are summarized in Appendix C and t-scores for two independent samples are included in Appendix D.

We measured students' perceived current knowledge of core concepts and methods of UXD taught in the class both with regards to themselves as individuals and to their teams. More significant increase is noticeable in the alternative design and evaluation phases of the UXD cycle. These pre-treatment scores prompted the instructor to re-visit those phases in the course design to enhance student learning. Also, evaluation was difficult without proper feedback from clients in the pre-treatment class.

Implementing the discussed active learning models with the related pedagogical approach showed significant improvement in students' perception of the extent to which the class structure, exercises, assessments, and projects meet their learning style. This emphasizes the demand for experiential learning and involving projects from the community.

6. LESSONS LEARNED

The class is subject to future enhancements but the instructor compiled a list of lessons learned from the re-designed class:

- Do not lose sight of industry demand and current practices. UXD is a dynamic area especially in the IS field and new techniques and tools emerge quickly.

- The project-based learning is a must in this class and makes a huge difference. Students were engaged and excited using a tool that helped them envision the enhanced design. Even a mock project makes a difference but having them see the impact their work made in their community is something they are proud of and motivated to do their best at.

- Provide enough time to do in-class work so students can ask questions or get help if they feel stuck. They provide higher quality work even if they have to finish later.

- The interviews and final presentations are essential in improving students' communication skills and confidence. The same goes with teamwork skills, which are mandatory in the current business climate.

- Seek end user input, even if not all stakeholders can provide feedback. Students seem to lose motivation without it.

- Do not let students assume they know what the users want. Even asking fellow students provides rich information from a different perspective.

7. REFERENCES

ACM/AIS Task Force. 2020. IS 2020 A Competency Model for Undergraduate Program in Information Systems. *Association for Computing Machinery,* Retrieved June 15, 2022 from https://www.acm.org/binaries/content/assets/education/curricula-recommendations/is2020.pdf

Abrahams, A. (2010). Creating e-Commerce Start-ups with Information Systems Students: Lessons Learned from New Venture Successes and Failures. *Information Systems Education Journal*, 8 (35).

Altay, B. (2014). User-centered design through learner-centered instruction. *Teaching in Higher Education*, 19(2), 138-155.

Benyon, D. (2019). Designing user experience: A guide to HCI, UX and interaction design (4th ed.). Harlow, England: Pearson.

Baham, C. (2019). Teaching Tip: Implementing Scrum Wholesale in the Classroom. *Journal of*

Information Systems Education, 30(3), 141-159.

Brenner, W., Osterle, H., & Petrie, C., et al. (2014). User, use & utility research: the digital user as new design perspective in business and information systems engineering. *Business Information Systems Engineering*, 6(1), 55–61.

Brubaker, E. R, Trego, M., Taha, K., Sheppard, S. D. (2018). A Closer Look at Campus-Community Partnerships in Engineering Education. IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, pp. 1-5, doi: 10.1109/FIE.2018.8658612.

Churchill, Bowser, & Preece (2013). Teaching and learning human-computer interaction: past present, and future. *Interactions,* 20(2), 44-53.

Cliburn, D. (2017). An interdisciplinary course on human-computer interaction. *Journal of Computing Sciences in Colleges*, 33(2), 38-45.

Computing Curricula 2020. (2020). *Association for Computing Machinery,* Retrieved on June 15, 2022 from https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf

Comeau, D. L., Palacios, N., Talley, C., Walker, E. R., Escoffery, C., Thompson, W. W., & Lang, D. L. (2019). Community-engaged learning in public health: An evaluation of utilization and value of student projects for community partners. *Pedagogy in Health Promotion*, 5(1), 3-13.

Connolly, T. M. & Begg, C. E. (2006). A Constructivist-Based Approach to Teaching Database Analysis and Design. *Journal of Information Systems Education,* 17(1), 43-54.

Dadarkar, V., Tiwari, A. (2022). Heuristics of Smartphones and Tablets to Identify Human Factors for Improving User Experience. In: Chakrabarti, D., Karmakar, S., Salve, U.R. (eds) Ergonomics for Design and Innovation. HWWE 2021. Lecture Notes in Networks and Systems, vol 391. Springer, Cham.

Dirin, A., Laine, T. (2018). User Experience in Mobile Augmented Reality: Emotions, Challenges, Opportunities and Best Practices. *Computers*, 7(2). https://doi.org/10.3390/computers7020033

Dix, Finlay, Abowd, & Beale. (2004). Human-Computer Interaction. (3rd ed.): Pearson.

Driscoll, M. (2000). Psychology of learning for instruction. Needham Heights, MA: Allyn & Bacon.

Hu, S., Kuh, G. D., & Li, S. (2008). The effects of engagement in inquiry-oriented activities on student learning and personal development. *Innovative Higher Education*, 33(2), 71-81.

Getto, G., Potts, L., Salvo, M. J., & Gossett, K. (2013). Teaching UX: Designing programs to train the next generation of UX experts. *In Proceedings of the 31st ACM Conference on the Design of Communication.*

Getto, G., & F. Beecher. (2016). Toward a Model of UX Education: Training UX Designers Within the Academy. *IEEE Transactions on Professional Communication*, 59(2), 153–64.

Gull, H., Saeed, S., Iqbal, S. Z., Saqib, M., Bamarouf, Y. A., & Alqahtani, M. A. (2018). Reflections on Teaching Human Computer Interaction Course to Undergraduate Students, *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 659-662.

Haaksma, T. R., de Jong, M. D. T., & Karreman, J. (2018). Users' Personal Conceptions of Usability and User Experience of Electronic and Software Products, in *IEEE Transactions on Professional Communication*, 61(2), 116-132.

Hein, G. E. (1991). Constructivist Learning Theory. Presented at CECA (International Committee of Museum Educators) Conference, Jerusalem Israel, 15-22 October 1991.

Khair, M. N., Skudai, M., Malaysia, J., Nabil, A. (2011). Problem-Based Learning (PBL) and Project-Based Learning (PBL) in Engineering Education: A Comparison. Proceedings of the IETEC'11 Conference, Kuala Lumpur, Malaysia

Knox, R. C. & Nair, R. W. (2021). Educational Effectiveness, Accreditation Content and Community Engagement of Practitioner Directed Environmental Capstone Classes. In Proceedings of 2021 ASEE Midwest Section Conference.

Kocielnik, R., Amershi, S., & Bennett, P. N. (2019). Will you accept an imperfect AI? Exploring designs for adjusting end-user expectations of ai systems. In Proceedings of the *2019 CHI Conference on Human Factors in Computing Systems*, 1-14.

Kolb, D. (1984). Experiential learning: Experience as the source of learning and development. Englewood Cliffs, NJ; Prentice-Hall.

Kreber, C. (2007). What's It Really All About? The Scholarship of Teaching and Learning as an Authentic Practice. *International Journal for the Scholarship of Teaching and Learning*: 1(1), Article 3.

Larson, S., & Harrington, M.C.R. (2012). A survey of ABET accredited information systems undergraduate programs in the USA. In Proceedings of the Information Systems Educators Conference, ISSN: 2167–1435,v29, n1961, New Orleans Louisiana, USA.

Lauer, C., & Brumberger, E. (2016). Technical Communication as User Experience in a Broadening Industry Landscape, *Technical Communication*, 63(3), 248–64.

MacDonald, C. M., Rose, E. J., & Putnam, C. (2022). An industry in flux: Where does UX go from here? *Interactions*, 29(2), 54–58.

McCarthy, M., Grabowski, B., & Koszalka, T., (1998). Web-based Instruction and Learning: Analysis and Needs Assessment, NASA Technical Publication 1998-206547, NASA Center for Aerospace Information, Hanover, MD.

McCarthy, J., & Wright, P. (2004). *Technology as experience*. MIT Press.

Mulder, I. (2015). A pedagogical framework and a transdisciplinary design approach to innovate HCI education. *Interaction Design and Architecture(s) Journal,* 27, 115-128.

Payne, M., Flynn, J., Whitfield, J.M., (2010). Capstone Business Course Assessment: Exploring Student Readiness Perspectives. *Journal of Education for Business*. 83(3).

Preiser-Houy, L., Navarrete, C. (2011). A Community-Based Research Approach to Develop an Educational Web Portal *Information Systems Education Journal*, 9(1) 4-13.

Rakes, G. (1996). Using the Internet as a tool in resource-based learning environment. *Educational Technology*, 6 (2), 52-29.

Roblyer, M.D., Edwards, J. & Havriluk, M.A. (1996) Learning Theories and Integration Models (Chapter 3). In Roblyer, Edwards, & Havriluk, Integrating educational technology into teaching. Prentice Hall.

Smith, R. (2021). Developing a senior project management capstone course for information systems and technology majors. Issues in Information Systems, 22(3).

Sousa Santos, B. (2006). An introductory course on human-computer interaction: Programme, bibliography, practical classes and assignments. *Computers and Graphics*, 30, 658-668.

Spielhofer, T., & Haselberger, D. (2021, October). Framing self-organized online team collaboration in a higher education course on Informatics and Society. In *2021 IEEE Frontiers in Education Conference* (FIE) 1-9.

Rose, D. Meyer, A., Hitchcock, C., (2005). The Universally Designed Classroom: Accessible Curriculum and Digital Technologies. Harvard Education Press, Cambridge.

Shumba, R. (2006). The development of a Human Computer Interaction course at a senior synthesis course. *ACM SIGCSE Bulletin*, 38(2), 102-104.

Talone, A., Basavaraj, P., & Wisniewski, P.J. (2017). Enhancing human-computer interaction and user experience education through a hybrid approach to experiential learning. SIGITE'17, October 4-7, Rochester, NY, USA, 83-88.

Wei, K., Siow, J., Burley, D., (2007). Implementing Service-Learning to the Information Systems and Technology Management Program: A Study of an Undergraduate Capstone Course. *Journal of Information Systems Education*, 18(1), 125-136.

Witman, P., Richlin, L., & Arboleda et al. (2007). The Status of the Scholarship of Teaching and Learning in the Disciplines. *Center* for Teaching Learning, & Scholarship at George Southern University, Statesboro, GA, USA

APPENDIX A
UXD in IS Curriculums

We randomly selected 40 universities among each of the following three size universities: large, medium, small universities with IS courses. We defined large universities as those with enrollments of 30,000 or more, medium universities as those with enrollments between 10,000 and 30,000, while small universities as those with enrollments less than 10,000. We focused on only those with IS curriculums and reviewed their curriculum to check for UXD path or course. If a UXD course existed, we reviewed the course description and syllabus if it was available. We used this information to identify the current penetration of UXD in IS curriculum and the methodology used to deliver this discipline.

UXD in Curriculum	University	Size	Name of Program	Name of Course or Major/Concentration
No	Texas A&M University	Large	Management and Information Systems	-
No	California State University, Fullerton	Large	Business Administration, Information Systems Concentration, B.A.	-
No	Ohio State University	Large	Fisher School of Business	-
No	Rutgers	Large	Management Information Systems Major	-
Yes	Rutgers*	Large	M.S. Digital Marketing	Special Topic: UX & UI in Digital Marketing
No	Indiana University Bloomington	Large	Information Systems BSB	-
Yes	Temple	Large	Management Information Systems Major	User Experience Design
No	Florida State University	Large	B.S. DEGREE IN MANAGEMENT INFORMATION SYSTEMS	-
Yes	University of Georgia	Large	Management of Information System	User Experience Strategy
No	Georgia Institute of Technology	Large	Information Technology Management Concentration	-
No	New York University	Large	BS in Business, Technology and Entrepreneurship	-
Yes	Columbia University*	Large	MBA – Marketing Division	Intro to User Experience

No	University of California	Large	Berkeley – Business Administration	-
Yes	University of Southern California	Large	Arts, Technology and the Business of Innovation (BS) – Designing for Digital Experiences Minor	Designing Digital Experiences
Yes	University of Michigan	Large	BACHELOR OF SCIENCE IN INFORMATION	User Experience Design (UX) path
No	University of Florida	Large	BACHELOR OF SCIENCE IN BUSINESS ADMINISTRATION INFORMATION SYSTEMS	-
Yes	Boston University*	Large	INFORMATION SYSTEMS GRADUATE LEVEL	Human Centered Design
Yes	University of Wisconsin*	Large	MAD UX Certificate Program	
Yes	University of Wisconsin*	Large	Information MS	User Experience Design
No	University of Illinois	Large	Bachelor’s in Information Systems	-
No	Pennsylvania State University	Large	B.S. Management Information Systems	-
No	Purdue University	Large	Business Analytics & Information Management, BS	-
No	University of Washington	Large	IS Major	-
No	University of Connecticut	Large	Management Information Systems Major	-
No	University of Maryland	Large	Information Systems	-
No	University of Massachusetts	Large	Isenberg School of Management	-
No	University of Minnesota	Large	Management Information Systems B.S.B.	-
No	Virginia Tech	Large	Business Information Tech	-
Yes	Brigham Young University	Large	Experience Design and Management	Experience Design
No	University at Buffalo	Large	Bachelor of Science in Information Technology and Management	-
No	Michigan State University	Large	Information Tech Minor	-
Yes	North Carolina State University	Large	Information Technology Concentration	Utilize User Interface (UI) design
No	Auburn University	Large	Information Systems Management	-
No	University of Colorado Boulder	Large	Business School	-

No	University of South Carolina	Large	Business School	-
No	University of South Florida	Large	Business Analytics and Information Systems B.S.	-
No	University of Utah	Large	Information Systems	-
No	Arizona State University	Large	Business Technology Major	-
Yes	University of Arizona	Large	Information Science & Arts	ISTA 416 / INFO 516: Introduction to Human Computer Interaction
Yes	University of Houston	Large	Management Information Systems	DIGM 1376 – User Experience (UX) Principles
Yes	Texas Austin / School of Information	Large	Bachelor's Degree in Informatics	User Experience (UX) Design
Yes	American University	Medium	Information Systems and Technology	CSC-535 User Interface Analysis and Design (3)
No	Appalachian State University	Medium	Business, Finance and Information Technology Education	-
No	Baylor University	Medium	Information Systems and Analytics	-
No	Boston College	Medium	Information Systems Concentration	-
Yes	Carnegie Mellon University	Medium	IS Bs	User Experience (UX) Design Concentration
No	Central Michigan University	Medium	Information Technology Information System	-
Yes	Cornell	Medium	Information Science	UX (USER EXPERIENCE) DESIGN
No	DePaul University	Medium	Management Information Systems B.S.B.	-
No	Drexel	Medium	Drexel's LeBow College of Business	-
No	East Carolina University	Medium	Management Information Systems	-
Yes	Emory University	Medium	INFORMATION SYSTEMS & OPERATIONS MANAGEMENT (ISOM)	ISOM 458 - User Experience Design (UXD)
No	Fordham University	Medium	Information Systems Program	-
No	Georgetown University	Medium	OPERATIONS & INFORMATION MANAGEMENT	-
Yes	Harvard University	Medium	Computer Science	User Experience Engineering
No	James Madison University	Medium	Computer Information System	-

No	Massachusetts Institute of Technology	Medium	Business Analytics	-
No	Metropolitan Community College	Medium	Information Technology	-
No	Mississippi State University	Medium	Management & Information Systems	-
Yes	New Jersey Institute of Technology	Medium	B.A. in Information Systems	User Experience Design
No	Northeastern University	Medium	Management Information Systems	-
No	Northern Arizona University	Medium	Information Systems	-
No	Oakland University	Medium	MANAGEMENT OF INFORMATION SYSTEMS	-
No	Ohio University	Medium	Business Management Technology Major	-
No	Old Dominion University	Medium	E-Business & E-Commerce (Information Systems & Technology, B.S.B.A.)	-
No	Rochester Institute of Technology	Medium	Management Information Systems	-
Yes	University of Kansas	Medium	Information Systems	Advanced Design Studies – Foundations in UI/UX Design
Yes	University of Miami	Medium	Business Technology Major	-
No	University of New Hampshire	Medium	BUSINESS ADMINISTRATION MAJOR: INFORMATION SYSTEMS AND BUSINESS ANALYTICS OPTION (B.S.)	-
No	University of Notre Dame	Medium	Business Technology Minor	-
No	University of Oregon	Medium	Lundquist College of Business	-
No	University of Pennsylvania	Medium	Management – Business Analytics Concentration	-
No	University of Pittsburgh	Medium	Business Information System	-
No	University of Rochester	Medium	Bachelor of Science in Business Information Systems Track	-
No	University of Rhode Island	Medium	BAI, Business Analytics and Intelligence	-
No	University of Tennessee	Medium	Accounting & Info Management	-
Yes	University of Texas–Dallas	Medium	Bachelor of Science in Information Technology and Systems	CS4352 – Human-Computer Interaction I
No	University of Vermont	Medium	The Grossman School of Business	
Yes	Utah State University	Medium	Information Systems Major	Web Development
No	Villanova University	Medium	MANAGEMENT INFORMATION SYSTEMS – MAJOR AND MINOR	-

No	Washington University in St. Louis	Medium	-	-
No	Capitol Technology University	Small	Bachelor of Science (BS) in Management of Cyber and Information Technology	-
No	Bellevue University	Small	BACHELOR OF SCIENCE Information Technology Degree	-
No	Brandeis University	Small	-	-
No	Bryant University	Small	INFORMATION SYSTEMS AND ANALYTICS DEPARTMENT	-
No	Butler University	Small	MANAGEMENT INFORMATION SYSTEMS	-
No	California Institute of Technology	Small	Information and Data Sciences Information Science and Technology	-
No	Chapman University	Small	-	
No	Clark University	Small	-	-
No	Clarkson University	Small	Information Systems	-
No	Colorado School of Mines	Small	BUSINESS ENGINEERING AND MANAGEMENT SCIENCE	-
No	Creighton University	Small	Information Technology	-
No	Drake University	Small	Information Systems	-
No	Gonzaga University	Small	Management Information Systems	-
No	Hamilton College	Small		
No	Illinois Institute of Technology	Small	Business Administration (B.S.)/Information Technology and Management (B.A.C.)	
No	Lehigh University	Small	Business Information Systems Major	-
No	Loyola Marymount University	Small	Information Systems and Business Analytics (ISBA)	-
Yes	Marquette University	Small	Information Systems	Business Applications Development
Yes	Michigan Technological University	Small	Management Information Systems	MIS 3500 – User-Centered Design
No	Misericordia University	Small	Information Technology	-
Yes	Missouri University of Science and Technology	Small	BUSINESS AND MANAGEMENT SYSTEMS	IS&T 1551 Implementing Information Systems: User Perspective
No	Pepperdine University	Small	-	-
No	Princeton University	Small		
No	Rensselaer Polytechnic Institute	Small	The management information systems concentration	-
No	Rice University	Small	-	-

No	Samford University	Small	-	-
No	Santa Clara University	Small	Information Systems and Analytics Department: Management Information Systems (MIS)	-
No	Seattle University	Small	-	-
No	Simmons University	Small	Information Technology	-
No	State University of New York College of Environmental Science and Forestry	Small		
No	Stevens Institute of Technology	Small	Information Systems	-
No	Texas Christian University	Small	Business Information Systems	-
Yes	Tuskegee University	Small	Information Technology (IT)	HUMAN-COMPUTER INTERACTION.
Yes	University of Dayton	Small	Management Information Systems	Introduction to Business Applications: Problem Solving with Visual Tools
No	University of La Verne	Small	Information Technology	-
No	University of Saint Joseph	Small	Business Intelligence & Analytics Major	-
No	University of San Diego	Small	Information Technology Management	-
No	University of Tulsa	Small	Computer Information Systems	-
No	Wake Forest University	Small	-	-
Yes	Worcester Polytechnic Institute	Small	Innovation with User Experience Certificate Program	UX Design

Some schools did not have IS curriculum but some related coursework.

*Graduate program

APPENDIX B
Student Feedback Survey Questions

Instructions: Please think through your learning journey in the class regarding user experience design. Please rate each question. Your answers are anonymous, they cannot be linked to you and the results will be reported in an aggregate level.

Question 1: Prior to this course, I was __________ of UXD principles and practices.

A	B	C	D	E
Not knowledgeable at all	Not very knowledgeable	Neutral	Somewhat knowledgeable	Very knowledgeable

Questions 2-7: Currently I have an adequate knowledge of ________.

2. Goal of user experience design (UXD).

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

3. Requirements gathering for UXD.

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

4. Presenting findings from requirements gathering.

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

5. Alternative designs

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

6. Prototyping

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

7. Evaluation

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

Questions 8-13: My team has an adequate knowledge of ________.

8. Goal of user experience design.

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

9. Requirements gathering for UXD.

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

10. Presenting findings from requirements gathering.

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

11. Alternative designs

A	B	C	D	E

Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

12. Prototyping

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

13. Evaluation

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

14. Overall, I am (now) knowledgeable of UXD principles and practices

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

Questions 15-18: The _________ fit my learning style.

15. course structure

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

16. assignments

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

17. teamwork

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

18. project

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

19. I feel that the exercises and project enhanced my knowledge of UXD principles.

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

20. I feel comfortable doing UXD related tasks at a future job.

A	B	C	D	E
Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree

APPENDIX C
Summary of Student Survey Results

Both prior and after the re-design values are included for the mean values by measurement items. The standard deviations are in parentheses. *P>0.05

Measures	**Prior** Mean (St. dev)	**After** Mean (St. dev)
Prior UXD knowledge Q1*	1.82 (0.62)	1.97 (0.71)
Current UXD knowledge Q2* Goal of UXD	4.52 (1.08)	4.71 (1.13)
Q3* Requirements gathering	4.44 (1.01)	4.68 (0.95)
Q4 Presenting findings	4.11 (0.84)	4.69 (0.94)
Q5 Alternative designs	3.91 (1.21)	4.41 (1.08)
Q6 Prototyping	4.09 (0.89)	4.52 (0.93)
Q7 Evaluation	3.88 (1.34)	4.49 (1.21)
Perceived team UXD knowledge Q8* Goal of UXD	4.55 (0.79)	4.78 (0.94)
Q9* Req. gathering	4.38 (1.12)	4.70 (1.03)
Q10* Pres. Findings	4.23 (1.10)	4.58 (1.03)
Q11 Alt. designs	3.81 (1.44)	4.52 (1.38)
Q12 Prototyping	4.21 (1.03)	4.89 (0.98)
Q13 Evaluation	3.79 (1.42)	4.54 (1.12)
Q14 Overall knowledge	4.11 (1.12)	4.68 (0.87)
Learning Style Fit Q15* course structure	4.02 (1.48)	4.38 (1.21)
Q16 assignments	3.73 (1.89)	4.31 (1.13)
Q17* teamwork	3.14 (1.72)	4.12 (1.18)
Q18 project	3.34 (1.62)	4.48 (1.41)
Future outlook – *I feel____* Q19 *… that the exercises and project enhanced my knowledge of UXD principles.*	4.12 (1.12)	4.73 (0.88)
Q20* *… comfortable doing UXD related tasks at a future job.*	3.12 (1.92)	3.82 (1.68)
N	32	55
Average grade-class	3.45 (0.52)	3.62 (0.48)

APPENDIX D
t-scores for two independent samples

The standard deviations are roughly equal, mean scores of answers to questions 1, 2, 3, 8, 9, 10, 15, 16, and 20 did not show statistically significant differences between prior and after redesign scores on the p<0.05 significance level. These questions were related to concept that students seemed to be the most comfortable with from other classes, for example, questions related to teamwork and requirements gathering. It suggests that new concepts specific to UXD benefited the most from the class redesign following the ACM recommended competencies embedded in the UXD course coupled with constructivist pedagogical model.

	prior redesign			after redesign															
Q#	**mean**	**sd**	**n**	**mean**	**sd**	**n**	s_p^2	**SE (x_1-x_2)**	**t statistic**	**DoF**	**> 0.2**	**0.2**	**0.1**	**0.05**	**0.02**	**0.01**	**0.005**	**0.002**	**0.001**
Q1	1.82	0.62	32	1.97	0.71	55	0.46	0.15	**0.99**	85	**x**	1.29	1.66	1.99	2.37	2.64	2.89	3.20	3.42
Q2	4.52	1.08	32	4.71	1.13	55	1.24	0.25	**0.77**	85	**x**	1.29	1.66	1.99	2.37	2.64	2.89	3.20	3.42
Q3	4.44	1.01	32	4.68	0.95	55	0.95	0.22	**1.11**	85	**x**	1.29	1.66	1.99	2.37	2.64	2.89	3.20	3.42
Q4	4.**11**	0.84	32	4.69	0.94	55	0.82	0.20	**2.88**	85		1.29	1.66	1.99	2.37	2.64	**2.89**	**3.20**	3.42
Q5	3.91	1.21	32	4.41	1.08	55	1.27	0.25	**1.99**	85		1.29	1.66	**1.99**	**2.37**	2.64	2.89	3.20	3.42
Q6	4.09	0.89	32	4.52	0.93	55	0.84	0.20	**2.11**	85		1.29	1.66	**1.99**	**2.37**	2.64	2.89	3.20	3.42
Q7	3.88	1.34	32	4.49	1.21	55	1.59	0.28	**2.18**	85		1.29	1.66	**1.99**	**2.37**	2.64	2.89	3.20	3.42
Q8	4.55	0.79	32	4.78	0.94	55	0.79	0.20	**1.16**	85	**x**	1.29	1.66	1.99	2.37	2.64	2.89	3.20	3.42
Q9	4.38	1.12	32	4.7	1.03	55	1.13	0.24	**1.35**	85		**1.29**	**1.66**	1.99	2.37	2.64	2.89	3.20	3.42
Q10	4.23	1.1	32	4.58	1.03	55	1.12	0.23	**1.49**	85		**1.29**	**1.66**	1.99	2.37	2.64	2.89	3.20	3.42
Q11	3.81	1.44	32	4.52	1.38	55	1.97	0.31	**2.28**	85		1.29	1.66	**1.99**	**2.37**	2.64	2.89	3.20	3.42
Q12	4.21	1.03	32	4.89	0.98	55	1.00	0.22	**3.06**	85		1.29	1.66	1.99	2.37	2.64	**2.89**	**3.20**	3.42
Q13	3.79	1.42	32	4.54	1.12	55	1.53	0.28	**2.73**	85		1.29	1.66	1.99	**2.37**	**2.64**	2.89	3.20	3.42
Q14	4.11	1.12	32	4.68	0.87	55	0.94	0.22	**2.65**	85		1.29	1.66	1.99	**2.37**	**2.64**	2.89	3.20	3.42
Q15	4.02	1.48	32	4.38	1.21	55	1.73	0.29	**1.23**	85	**x**	1.29	1.66	1.99	2.37	2.64	2.89	3.20	3.42
Q16	3.73	1.89	32	4.31	1.13	55	2.11	0.32	**1.79**	85		1.29	**1.66**	**1.99**	2.37	2.64	2.89	3.20	3.42
Q17	3.14	1.72	32	4.12	1.18	55	1.96	0.31	**3.15**	85		1.29	1.66	1.99	2.37	2.64	**2.89**	**3.20**	3.42
Q18	3.34	1.62	32	4.48	1.41	55	2.22	0.33	**3.44**	85		1.29	1.66	1.99	2.37	2.64	2.89	3.20	**3.42**
Q19	4.12	1.12	32	4.73	0.88	55	0.95	0.22	**2.82**	85		1.29	1.66	1.99	2.37	**2.64**	**2.89**	3.20	3.42
Q20	3.12	1.92	32	3.82	1.68	55	3.14	0.39	**1.78**	85		1.29	**1.66**	**1.99**	2.37	2.64	2.89	3.20	3.42

The distribution table followed DF=80 from https://www.medcalc.org/manual/t-distribution-table.php
The above analysis was provided by an anonymous reviewer.

Prospects of Autonomous Vehicle Learning Kits in Education Systems

Biju Bajracharya
bajracharya@etsu.edu

Mohammad Shoeb Khan
khanms@etsu.edu

Department of Computing
East Tennessee State University
Johnson City, TN 37614 USA

Abstract

The Autonomous Vehicles (AV) are a self-driving vehicle capable of sensing its environment and operating with minimal or no human intervention converting into a fully or partially automated vehicle. These automated vehicles have great potential to revolutionize the automotive industry and our daily lives. Thus, they are receiving a lot of attention from automotive industries, governments, suppliers, educational sectors, researchers, and many other stakeholders. According to global financial service firm UBS, the AV market could reach more than $2 trillion by 2030. These projections will become reality if enough skilled workforces are produced with a high level of broad technical competencies specifically for the automotive environment. This will also require consumer education on AVs that could bring all the comfort and excitement across all segments of the population. This brings a substantial opportunity for the educational sectors in generating the workforce required in AV sectors by enhancing the existing curricula in the areas related to AVs or developing a cross-disciplinary course or capstone project that could expose students with a technical background in computer science, computer engineering, electrical engineering, and mechatronics. Since AV has many technologies involved, it is challenging to design an appropriate curriculum and provide hands-on activities. Small-scale learning kits may be an affordable and effective solution for introductory courses on AVs or for beginners. This paper will discuss the use of those widely available small-scale learning kits, their benefits, and their challenges.

Keywords: Autonomous Vehicles, AV, Autonomous Systems, Artificial Intelligence, AI, Learning Kits, AV Education, IoT Application, IoT Development Kit, IoT Starter Kit.

1. INTRODUCTION

Automation and robotics technologies have traditionally been used by the manufacturing industries. Recent advancement of these technologies is spurting into an autonomous system that can respond to real-world conditions with minimal or no human intervention. In such an autonomous system, one can achieve a given set of goals in a changing environment—gathering information about the environment and working for an extended period without human control or intervention (Ultimate Guides, n.d.). These autonomous systems (AS) need the ability to acquire data about its environments and adapt or refine its behavior in real-time (MIT AeroAstro, 2021). AS is bringing enormous efficiencies in productivity, unmanned surveillance, unmanned navigation, and handling of harmful environments. It focuses on developing embodied intelligent systems, for example, intelligent navigation, remote monitoring, autonomous warehouse, autonomous drones, autonomous vehicles, etc. With so many potential application areas, it can have a strong economic contribution as an industrial and commercial activity and disruptive socio-economic impact across diverse market sectors worldwide (Robotics and Autonomous Systems, n.d.).

According to CIO analysts Kevin Dennean, Rolf Ganter, and Hartmut Issel,the AV industry could reach more than $2 trillion by 2030 (UBS, 2021). It will create more than 100,000 jobs in the AV industry and 30,000 jobs for engineers with computer science degrees that the US can't fill (Rayome, A. D. N., Staff, T. R., Fernandez, R., Okeke, F., Miles, B., Bohon, C., & Librescu, M., 2019). For this huge growth potential of AV sectors and workforce gap, it is required to generate a well-trained workforce equipped with the necessary skills that are required for conducting research and development projects in AV industries. This creates the opportunity for educators to effectively design and deliver related course content.

2. AV IN EDUCATION SYSTEMS

With the immense growth potential of AV sectors, many sectors including governments, transportation sectors, suppliers, traditional auto manufacturers, etc. are attracted to AV. Educational sectors and educators have an excellent opportunity to update their curricula to incorporate AVs in their educational activities. Since AV is a complex cross-disciplinary subject that has many technologies involved, it is challenging to design an appropriate education and training program. Rather, most of the existing courses focus on one or two technologies (Liu, S., Gaudiot, J.-L., & Kasahara, H., 2021). The major issue is what should be emphasized: hardware, software, programming languages, artificial intelligence, car platforms, tools, software development frameworks, computing platforms, architectures, or other topics (Bastiaan, J., Peters, D., Pimentel, J., & Zadeh, M., 2019). The most important change in education will be emphasizing the electrical/electronic systems and computer diagnostics components (Hadfield, C., 2020).

There are three major categories of AV in education activities:

a) Integrating AV topics on the Existing Curriculum for example Control Systems, Computer Vision, AI, Automation, Image Recognition, smart cities, Intelligent Transportation Systems, Machine Learning, etc.
b) Incorporating AV courses in the New Curriculum for example: Control of Autonomous Vehicles, Vehicle Dynamics, Vehicular Communication, Ethics and Legal Issues
c) Offering Project Based Course related to Autonomous Vehicles topics

3. AV LEARNING KITS/TOOLS

AV learning kits or tools consist of a set of relatively inexpensive and flexible components, including breadboards, jumper wires, development controller boards, sensors, motors, driving mats, and electric components like resistors, capacitors, and inductors. Commonly used development boards are Arduino-UNO, Raspberry Pi, Intel Galileo boards, beagle bone, Jetson Nano, etc. These kinds of kits are called do-it-yourself kits (DIY Kits) or ready-to-run (R2R) kits.

These kinds of DIY learning kits come as all-in-one packet kits along with all the required components and step-by-step instructions (Level 5 Supplies n.d.). Some kits may require additional components or a robot car which will be included in the DIY instruction sets. Most of these kits come with a smartphone app or computer software that allows easy connection with preloaded projects. Some kits are especially designed for STEM high school students (Dextered, n.d., James, 2022, Home, n.d.). Example of such DIY kits are:

a) DIY Robocars
b) Duckietown
c) Formula Pi
d) Udacity Self Driving Car
e) Elcano Project
f) JetBot AI Kit
g) Robolink Zumi AI Self-Driving Car Kit
h) Picar-X

Figure 1: DIY 4WD Robotic Car: Robotics Learning Kit (Robot Pi Shop, n.d.)

AV KITS PLATFORMS

AV kits consist of hardware, software, cloud, and communication platforms. The accessories like chassis, wheels, battery holders, jumper wires, etc., are used to assemble the kit. Hardware platforms use the programmable circuit board also known as a microcontroller board. These boards receive the programs from the computer through an integrated development environment (IDE). IDE is the graphical user interface used to write a program and upload it to the board. These controller boards can be expanded using the expansion boards or breakout boards for additional features and functionalities or for easier connection and assembly of the kit. For example, if the hardware platform is Arduino Microcontroller, then expansion boards are called shields. These shields can be a motor shield, relay shield, LCD shield, ethernet shield, proto shield, CAN-BUS shield, smoke detector shield, GSM/GPRS shield, camera shield, etc. These shields add additional features to the board like ethernet, Bluetooth, camera, smoke detector, etc. The addition of several features and functions can demand higher processing power on the controller board. To support higher processing power on board, an edge control board can be used. Edge control boards support the deployment of artificial intelligence (AI) and machine learning on the board (edge) called AI edge processing. However, these kits can be limited by the budget constraints on the education kits.

The software platform is used to write customized code that can be uploaded to the controller board. These codes can be written in Java, Python, C, etc. Some vendors also provide prewritten code and tutorials. To control the board remotely, or to transfer and store the data, a cloud platform is used. Cloud or remote connection platforms use computer protocols like MQTT, CoAPP, http, etc. Communicating with the remote system requires network connectivity. This connectivity is provided by the network platform like Wi-Fi, Bluetooth, Zigbee, Z-wave, etc. These platforms are summarized in the Table 1.

AV KITS FUNCTIONS, FEATURES, AND USES

These DIY kits can support several functions, including object detection and recognition, cliff detection, lane following, path planning, etc. These functions generally divided into five categories: computer vision, sensor fusion, localization, planning, and control. To support these functions, it uses sensors. Sensors are made into ready-to-mount components called modules. The modules are available for all varieties of sensors. Among them, the camera module, ultrasonic sensor, and line tracking sensor are used to detect the color, obstacles, road signs, etc. The detection module are used to follow the track and react to the obstacles and the road signs. It can be used to detect the cliff so that it won't fall from the slopes. GPS module and compass module can be used to get the best position estimate of the robot kit.

These development kits are suitable for entry-level learners of AV or enthusiasts. These learners' groups can be STEM students, undergraduate students, or graduate students including research students. These kits can be further expanded by adding customized components. The platform, Duckietown, expands the kit with a customized list of components and parts to provide a most in-depth real-world experience of AVs. This platform is designed to support a wide range of functions at a low cost. Those functions are following lanes while avoiding obstacles, pedestrians (duckies), and other Duckiebots, localizing within a global map, navigating a city, and coordinating with other Duckiebots to avoid collisions (Paull, et. al., 2017).

These kits with expanded functions and features at the software or hardware levels, can be implemented or converted into real/practical world scenarios for example in detecting land mines, explosives etc.

Figure 2: Coimbra's Mine Detection Robot (Hennessey, M., 2015)

Some of the kits are designed for academic research for example Quanser's self-driving car research studio for the QCar. QCar is an open-architecture scaled model vehicle, powered with NVDIA® ® Jetson™ TX2 supercomputer, and equipped with a wide range of sensors, cameras, encoders, and user-expandable (Quanser, 2022).

Figure 3: The Self-Driving Car Research Studio and QCar (TecSolutions, Inc., n.d.)

The enhanced feature of the kits like QCar have on-board computing (OBC) chip. These on-board computing chip also known as Artificial Intelligence (AI) Edge Chip or Edge Control boards. These chips/boards deployment of artificial intelligence (AI) and machine learning on the board (edge) called AI edge processing.

These learning kits enables learning for various levels of learners. These learning process can be reinforced by participating in different kinds of AV or AI enhanced self-driving car leagues and competitions. Some of the leagues are listed in the Table 2.

Leagues	Descriptions
AI Driving Olympics	Held annually at the NeurIPS conference
Amazon DeepRacer League	Train and race low-cost ($300) Amazon 1/18th scale cars using AWS reinforcement learning
Self Racing Cars	San Francisco Area full-size cars and DIY Robocars events
Self Driving Track Days	Europe: test track and one-day training courses
F1/10 Racing	higher-end 1/10th scale cars with Lidar and Nvidia computer, started by U Penn. Starting cost around $3,000

Table 2: Self Driving Car Leagues and Competitions

4. BENEFITS AND FUTURE TRENDS

There are several benefits of adopting Learning AV Kits. Some of these benefits are as follows:

- There are many DIY or R2R kits available in the market based on the need and affordability.
- These kits are available for different age groups K12 to college students.
- These kits are suitable for any skill level from beginners to advanced.
- It does not require prior skills or knowledge. The kits come with step-by-step instructions set, and online tutorials or YouTube videos are also available.
- Most of the Kits come with cloud application integrations.

AV industries will create new jobs while other traditional vehicle industries will be greatly impacted which could cause a loss of more than four and a half million jobs worth $168 billion dollars of annual wages (UPCEA, 2017). In a study by the Institute of Transportation and Development Policy in 2017, there are three trends that, if adopted concurrently, would unleash the full potential of autonomous cars: vehicle automation, vehicle electrification, and ridesharing. By 2050, these "three revolutions in urban transportation" could (Synopsys, n.d.):

- Reduce traffic congestion (30% fewer vehicles on the road)
- Cut transportation costs by 40% (in terms of vehicles, fuel, and infrastructure)
- Improve walkability and livability
- Free up parking lots for other uses (schools, parks, community centers)
- Reduce urban CO_2 emissions by 80% worldwide

According to McKinsey & Company, there are three-time horizons of AV diffusion before such vehicles become commercially available to individual buyers when they are in the early stage of adoption, and when they become the primary means of transport illustrated in Figure 4. It is highly likely that jobs created related to AVs will also grow tremendously during these time horizons.

However, there are some variations among the researchers or agencies regarding the projected data in the next few decades, all those projections have high growth projections. From these, it is evident that AV projection in the next few decades will be an indispensable part of the modern transport system. To develop skilled manpower that can match with growth rate, it is urgent for the educational community to reform their curricula to ensure they cover diverse subjects that are required for AVs including software engineering, communication technologies, electrical engineering, etc (Bagloee, 2016, Chaudhury et al, 2016).

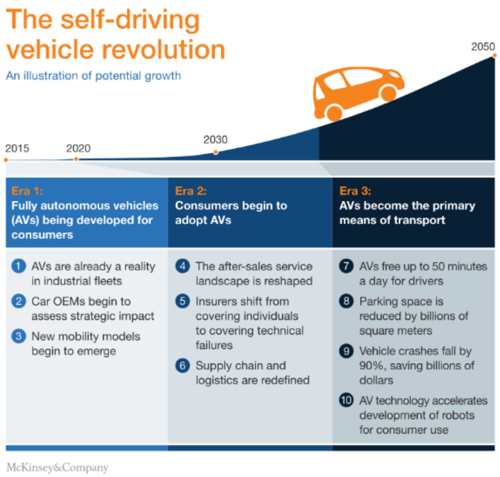


Figure 4: The Self-Driving Vehicle Revolution and the Three Eras of AVs (Bertoncello, M., & Wee, D., 2021)

5. CHALLENGES OF EDUCATIONAL IoT KIT

Currently, AVs are one of the major innovations being one of the areas with greatest potential for transformation of society and the economy in the coming decades. As such, all the involved stakeholders including developers, industries, educators, and consumers, face several challenges that remain to be tackled. Some of the challenges in the education system are as follows:

- Educators also have the challenge of adapting AVs into programs and curricula at a rapid pace to match the pace of adoption of the technology in the real world (Hadfield, C., 2020).
- Need to train students the individual components and sub-systems like lane following, road markings detection, obstruction detection, coding, etc. which may not fit a single course
- Educators' reluctance to adopt new technology.
- Educators may need to partner with employers/vendors to perform outreach to the multitude of careers with autonomous technology (Hadfield, C., 2020).
- There are many available DIY kits available in the market and it may be hard for the beginner to know which one to choose.
- The cost of the kits may range from $200 to $1000s. These costs may not be affordable for some students.
- Troubleshooting the hardware components could be difficult to diagnose for the beginner.
- Some devices and applications may not be compatible making them difficult to deploy.
- Students will learn the basic concepts of how autonomous vehicle works from the small-scale learning kits; however, the complexity of a real on-road autonomous vehicle is much greater and more robust.

6. CONCLUSION

Autonomous vehicle technology is on the horizon and widespread adoption of these technologies will grow rapidly over the next few decades as discussed in the self-driving vehicle revolution eras. These technologies will impact many sectors including government, auto industries, researchers, and educators, prompting them to seek new avenues of development and innovations and ways of adapting to these technologies. Educators need to incorporate or blend these topics into their existing courses. Small-scale learning kits could be the cost-effective solution that can speed up the learning process and the realization of the potential of AVs can soon be the profound reality of driverless cars in the smart world.

7. REFERENCES

Accelerate your research with Quanser's self-driving car studio. Quanser. (2022, November 28). Retrieved January 02, 2023, from https://www.quanser.com/products/self-driving-car-studio/

Autonomous Systems: Ultimate guides: Blackberry QNX. Autonomous Systems | Ultimate Guides | BlackBerry QNX. (n.d.). Retrieved July 4, 2022, from https://blackberry.qnx.com/en/ultimate-guides/autonomous-systems

Autonomous Systems & decision-making. MIT AeroAstro. (2021, September 21). Retrieved July 4, 2022, from https://aeroastro.mit.edu/research-areas/autonomous-systems-decision-making/

Bagloee, S. A., Tavana, M., Asadi, M., & Oliver, T. (2016). Autonomous vehicles: Challenges, opportunities, and future implications for transportation policies. Journal of Modern Transportation, 24(4), 284–303. https://doi.org/10.1007/s40534-016-0117-3

Bastiaan, J., Peters, D., Pimentel, J., & Zadeh, M. (n.d.). The AutoDrive Challenge: Autonomous Vehicles Education and training issues. 2019 ASEE Annual Conference &

Exposition Proceedings. https://doi.org/10.18260/1-2--33371

Bertoncello, M., & Wee, D. (2021, June 22). Ten ways autonomous driving could redefine the Automotive World. McKinsey & Company. Retrieved July 5, 2022, from https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/ten-ways-autonomous-driving-could-redefine-the-automotive-world

Chatzopoulos, A., Papoutsidakis, M., Kalogiannakis, M., & Psycharis, S. (2020). Innovative Robot for Educational Robotics and STEM. Intelligent Tutoring Systems, 95–104. https://doi.org/10.1007/978-3-030-49663-0_13

Chowdhury, M., & Dey, K. (2016). Intelligent Transportation Systems-a frontier for breaking boundaries of traditional academic engineering disciplines [education]. IEEE Intelligent Transportation Systems Magazine, 8(1), 4–8. https://doi.org/10.1109/mits.2015.2503199

Dextered - educational robots with the Raspberry Pi. Dexter Industries. (n.d.). Retrieved September 10, 2022, from https://www.dexterindustries.com/dextered/

Hadfield, C. (2020, September 21). Autonomous Vehicle Technologies: The impact on workforce training and education. Minnesota State Transportation Center of Excellence. Retrieved July 5, 2022, from https://www.minntran.org/autonomous-vehicle-technologies-and-the-impact-on-workforce-training-and-education/

Hennessey, M. (2015, September 28). ICRA to host challenge for Landmine Detection. Clearpath Robotics. Retrieved Dec 25, 2022, from https://clearpathrobotics.com/blog/2014/04/icra-challenge-landmine-detection/

Home - VEX robotics. Home - VEX Robotics. (n.d.). Retrieved September 16, 2022, from https://www.vexrobotics.com/

James. (2022, July 18). Best Robotic Kits for high school students. STEM Toys UK. Retrieved September 16, 2022, from https://www.stemtoys.uk/buyers-guides/best-robotic-kits-for-high-school-students/

Liu, S., Gaudiot, J.-L., & Kasahara, H. (2021). Engineering education in the age of Autonomous Machines. Computer, 54(4), 66–69. https://doi.org/10.1109/mc.2021.3057407

Make fun arduino robot 4WD car. Robot Pi Shop. (n.d.). Retrieved January 6, 2023, from https://robotpishop.com/make-fun-arduino-robot-4wd-car.html

Paull, L., Tani, J., Ahn, H., Alonso-Mora, J., Carlone, L., Cap, M., Chen, Y. F., Choi, C., Dusek, J., Fang, Y., Hoehener, D., Liu, S.-Y., Novitzky, M., Okuyama, I. F., Pazis, J., Rosman, G., Varricchio, V., Wang, H.-C., Yershov, D., … Censi, A. (2017). Duckietown: An open, inexpensive and flexible platform for Autonomy Education and research. 2017 IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/icra.2017.7989179

Rayome, A. D. N., Staff, T. R., Fernandez, R., Okeke, F., Miles, B., Bohon, C., & Librescu, M. (2019, November 4). Self-driving cars will create 30,000 engineering jobs that the US can't fill. TechRepublic. Retrieved July 5, 2022, from https://www.techrepublic.com/article/self-driving-cars-will-create-30000-engineering-jobs-that-the-us-cant-fill/

Robotics an Autonomous Systems. Joinup. (n.d.). Retrieved July 4, 2022, from https://joinup.ec.europa.eu/collection/rolling-plan-ict-standardisation/robotics-autonomous-systems

Self-driving car research studio. TecSolutions, Inc. (n.d.). Retrieved January 5, 2023, from https://tecsolutions.us/quanser/product/self-driving-car-research-studio

Tang, J., Shaoshan, L., Pei, S., Zuckerman, S., Chen, L., Shi, W., & Gaudiot, J.-L. (2018). Teaching autonomous driving using a modular and integrated approach. 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). https://doi.org/10.1109/compsac.2018.00057

The effect of autonomous vehicles on education. UPCEA. (2017, November 6). Retrieved July 4, 2022, from https://upcea.edu/autonomous-vehicles/

Three revolutions in Urban Transportation. Institute for Transportation and Development Policy. (2017, May 3). Retrieved July 15, 2022, from https://www.itdp.org/2017/05/03/3rs-in-urban-transport/

Top 5 autonomous car learning tools. Level Five Supplies. (n.d.). Retrieved July 5, 2022, from https://levelfivesupplies.com/top-5-autonomous-car-learning-tools/

UBS. (2021, February 25). Electrified vehicles could make up 60-70% of global sales by 2030. Retrieved July 5, 2022, from https://www.ubs.com/us/en/wealth-management/insights/market-news/article.1562721.html

What is an autonomous car? – how self-driving cars work. Synopsys. (n.d.). Retrieved July 2, 2022, from https://www.synopsys.com/automotive/what-is-autonomous-car.html

Developing a Data-Driven Emerging Skill Network Analytics Framework for Automated Employment Advert Evaluation

Xiaoming Liu
xliu@semo.edu
Department of Computer Science

Dana Schwieger
Department of Management
dschwieger@semo.edu

Southeast Missouri State University
Cape Girardeau, MO 63701, USA

Abstract

Rapid advancements and emergent technologies add an additional layer of complexity to preparing computer science and information technology higher education students for entering the post pandemic job market. Knowing and predicting employers' technical skill needs is essential for shaping curriculum development to address the emergent skill gap. Examining online advertisements to determine the skills sought by employers of new hires for these emerging areas and ensuring that program course content addresses these skills can be a daunting task. In this paper, the authors describe the development of a data-driven analytics framework that can be used for evaluating emerging skill clusters in online job adverts and the application of the framework to a mobile computing course at the authors' institution.

Keywords: Graph mining, network clustering, time-evolving network, emerging technology skills analysis framework

1. INTRODUCTION

In light of the current technology knowledge explosion, rapid advancements and continuous innovations require various actors (students, employers, educators, etc.) to quickly adapt in order to stand at the forefront of the competitive edge (Mitchel, 2022; Sun et al., 2021). In order to fill the "Skill Gap" left in the wake of these advancements, information systems (IS) educators must be able to understand and incorporate emerging information technology (IT) innovations to prepare students for future success (Agarwal & Ahmed, 2017; Liu & Murphy, 2012; Mitchel, 2022). Meeting industry demand for such highly dynamic technical skills provides a significant challenge for IS educators (Leidig et al., 2020; Mitchell, 2022). The accelerated digital transformation of higher education institutions associated with new online platforms, tools, and teaching modalities wrought by the COVID-19 epidemic adds additional challenges to effectively evaluating current online learning content and meeting the needs of the job market (Alenezi, 2021).

In recent years, online job advert-derived analytics solutions for skill demand assessments have been developed and implemented (Sun et al., 2021; Tamašauskaitė & Groth, 2022). Likewise, researchers have adopted data-driven analytics approaches to assessing IS/IT courses and curriculum (de Blas et al., 2021; Yu et al., 2021). Nevertheless, a quantitative framework is lacking for effectively assessing the emerging technology-related learning content in IS/IT courses through combining up-to-date and more accessible online job advert analytics with online course content analytics.

In this paper, the authors propose a data-driven emerging skills analytics framework combining online job advert analytics with online course content analytics for automated knowledge interaction evaluation in IS/IT courses. Through applying the proposed analytics framework, students are able to maximize their skill value in the job market. IS/IT educators are also able to provide adaptive and up-to-date learning content relative to the current job market demand. Furthermore, employers may use the skill-centric skill assessment to recruit and retain skilled talents. The proposed framework consists of a conceptual university-industry knowledge interaction model, online job skill network analytics module, and an online course content analytics module for automated knowledge interaction evaluation in IS/IT curriculum. More specifically, the framework extracts graphlets with local-topological statistics from generated skill networks for role-based skill interaction analysis. A case study in an online mobile application development course was implemented for proof of concept and early verification.

2. RELATED WORKS

Identifying information technology job skills sought by employers has long been of interest to job seekers, academes, human resource administrators, and many others (Cummings & Janicki, 2020; Koong, Liu, & Liu, 2002; Morris, Fustos, & Haga, 2018). Research has been conducted using data mining and valuation models to identify and value skills in online job adverts (Sibarani & Scerri, 2019; Smith & Azad, 2014). Further, technology innovations and the related emergent skill sets are fostered by the symbiotic relationship between education, research, and industry. To address the rapid evolution of technology development in the industrial space, it is imperative that universities develop support practices for gaining insights.

2.1 Knowledge Interaction

Cowan and Jonard (2001) noted that "…there are two aspects to technical change: knowledge creation and knowledge diffusion" (p. 328). They recognized the value of networks in the knowledge creation and diffusion process as both collective successes and failures are shared, processed, and examined by members of the sharing network. In his study of network epistemology, Zollman (2013) acknowledged the value generated by groups in knowledge development and transfer. His study focused on the particular nuances of the communication processes within these networks. Wijesinghi (2022) described the importance of collaborative relationships between educational institutions, research and development entities, industrial players, and innovation intermediaries (e.g., fellows, incubators, development agencies, etc.) in the development and transfer of technology innovations. Such relationships are essential to expeditious development and transfer of tacit, implicit, and explicit knowledge. da Silveria Bueno, et al. (2021) observed that global developments in the bioenergy field have been fueled by such collaborative networks. The authors noted that "…knowledge flows from the emerging networks and their relationships are outlining the frontier technologies in the bioenergy paradigm" (p. 15). Advancements in bioenergy production and other scientific fields illustrate how invaluable interdependent networks are to dynamically developing fields, technical organizations, and programs of study wishing to stay at the forefront of their fields.

2.2 Skill Network Analysis

Researchers have taken multiple approaches to addressing the volume and variety of network data. de Blas, et al., (2021) proposed the use of network analysis and dependency graphs in the design and development of undergraduate curriculum to reflect the temporal sequencing and dependencies of course content and its acquisition. The authors indicated that identifying key nodes and their relationships was one of the most important issues in the process.

To address the widening skill gap found in the European data economy between supply and demand, Sibarani and Scerri (2019) used hierarchical clustering with co-word occurrence to identify job skill advert network demand and composition. They hypothesized that "…skill demand can, to an extent, be discovered and predicted, by tracking the skills network evolution over a series of observances derived from web-posted job adverts" (p. 2). The authors assigned weights to the connections between clusters in the evaluation of relationships.

Espejo et al. (2020) noted that the evolution and behavior of networks could be understood by analyzing the topology of complex networks. The authors proposed using the GHuST framework to decompose muti-node networks of various sizes (e.g., Facebook, retweets, the Web, etc.) into 2- and 3- node graphlets for more manageable analysis and comparison.

Dadzie, et al. (2018) recognized the limitations of the human mind in meaningfully identifying trends in big data sets. The authors utilized task- and context-driven scenarios along with interactive graphic visualization techniques to iteratively explore and discover job skill demand trends and co-occurrences provided by job adverts found on online job boards. The graphics allowed users to visually identify the skills in most predominant demand.

2.3 Job Skill Analysis

In their work on examining job advert skill clusters, Sibarai and Scerri (2020) described a Skills Cluster Observation and Discovery (SCODIS) framework they used to develop a forecasting model for evolving skill networks to predict future high-demand and emerging skillsets. Sun, et al., (2021) developed a Salary-Skill Composition Network (SSCN) to extract job skills and measure their value based on immense job postings. The authors proposed a valuation model that was able to assign meaningful value to job skills in various contexts outperforming other models in predicting job salaries. The authors also suggested multiple applications for their model including the valuation of skills in the marketplace, salary predictions, knowledge and talent development, and guidance for job seekers. Such models can be used to address some of the current challenges faced by educators in that, with so many new technologies being introduced in industry, how do the emerging skills impact the current computing courses with dynamic job skill demand in high-tech industries?

3. ADDRESSING THE CURRENT CHALLENGE

In this paper, the authors draw upon the works of Dadzie et al. (2018), Espejo et al. (2020), Sibarai and Scerri (2020), Sun et al. (2021) to develop a data driven network analytics framework to compare course content with emerging skill sets sought in online job adverts. The framework can be used to:

- identify key industry workplace competencies to develop meaningful computing course content;
- provide an analytics framework for capturing future technology trends;
- support the development of interactive data driven analytics tools for discovering curricular opportunities; and
- provide insights for fostering information and communications technology (ICT) innovations in university-industry collaborative networks.

4. THE PROPOSED ANALYTICS FRAMEWORK

The proposed framework consists of one high-level conceptual model and two analytics modules for emerging technology-centric knowledge interaction evaluation. The conceptual university-industry knowledge interaction model (Fig. 1) was created to provide a complete picture of the solution and present the key elements and their interactions in an analytics framework. The emerging technology-centric job skills network analytics module (Fig. 2) was implemented for evaluating the job market demand of the emerging technology skills related to conventional skills. The online course content analytics module (Fig. 4) was implemented for evaluating the contextual information of the emerging technologies in a conventional computing course (Clear & Parrish, 2020, p. 189; Wijesinghi, 2022, p. 56). The details are presented in the following sections.

4.1 Conceptual University-Industry Knowledge Interaction Model

The conceptual university-industry knowledge interaction model, as depicted in Figure 1, presents the agents (students, educators, and employers) in a knowledge-based society and their interactions. (Each of the three agents are represented by a node in the triangles.) If the job market demand of interconnected conventional skills and emergent skills can be measured, the educator develops tailored cross-disciplinary teaching and hands-on learning content to support the adaptive learning environment and fill the identified gaps between the emerging technology skills demanded by industry and the current computing curricula. The proposed conceptual model provides a high-level picture of an agent-based skill network with university-industry interactions with emphasis on the fact that the process is iterative.

The area to the right of the dashed line represents industry influences and the changing skill set demands of employers. The area to the left of the diagonal dashed line in the model represents the interactions and transformations taking place at the university level between Educators as Stimulators of learning and Students as Actors in the learning process.

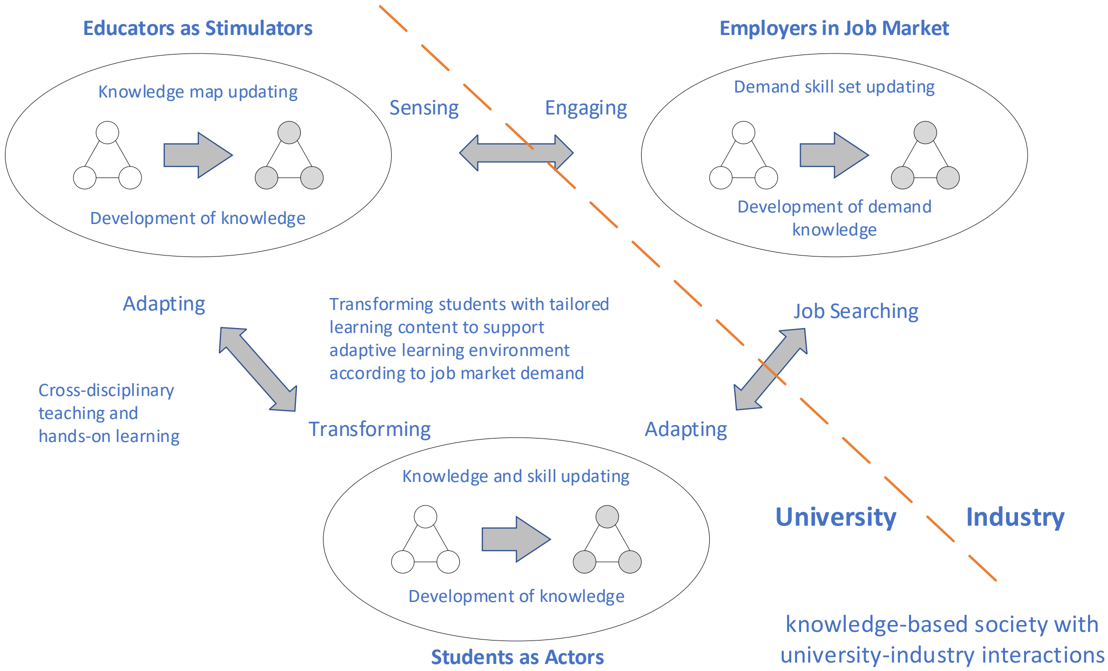


Figure 1: Conceptual University-Industry Knowledge Interaction Model

The arrows on the diagonal dashed line represent the interactions and influences that occur between industry and education (faculty and students) through the process of knowledge sharing, development, and growth. The small noded triangle sets on each side of the diagonal line represent the iterative process of changing states for each of the participants in the interaction process. For instance, as the faculty members realize the demand for particular skills in industry, they modify the content and their approach to teaching courses. As students learn the material, their skill set background enlarges, preparing them for further development. As graduates enter the industrial workforce prepared to address new and emerging technologies, industrial actors' demand for more technical skills is further heightened. Influences in the interrelationships in the iterative process may be driven by any of the actors (e.g., Educators as Stimulators, Students as Actors, Employers in the Job Market) in each of the cycles. Thus, the model recognizes the iterative nature of the symbiotic relationships between the actors.

4.2 Emerging Job Skill Network Analytics Module

This section includes an extensive overview of the emerging job skill network analytics module with online-posted job adverts (Fig. 2). More specifically, the authors focus on industry-demanded conventional skills having emerging characteristics. The online job advert dataset is represented as an undirected and weighted graph with a topology of interconnected skills and weight indices representing the associated strength between skills based on their observed job advert co-occurrences. The skill networks are then decomposed into six 2- and 3-node graphlets representing sets of highly interconnected conventional skills and emerging skills (Fig. 3). Through analyzing the skill graphlets, a quantitative result for evaluating job market demand for the emerging skills associated with a specific conventional skill can be provided. The role-based skill association strength is also calculated for the essential skill in leading and supporting roles based on the skill graphlets. The proposed framework, as depicted in Figure 2, comprises the steps that follow.

Step 1: Job advert dataset generation
A python program was developed to build a pipeline for job advert dataset generation (Dadzie et al., 2018).

Step 2: Skill interaction identification
Given a job advert with a job title string and a job context string, the skill interactions can be identified if the conventional skills and emerging skills appear in the title string and/or the context string. If the conventional skill set (C) appears in the title string and the emerging skill set (M) appears in the context string, every conventional skill is connected to every emerging skill (a complete bipartite graph) and the job advert has $C \times M$ skill interactions.

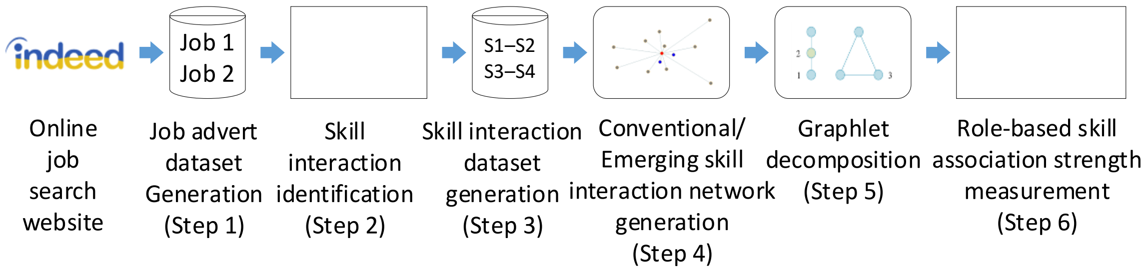


Figure 2: Emerging Job Skill Network Analytics Module

If the conventional and emerging skills only appear in the context string, the conventional/emerging skills are fully interconnected (a complete graph) and the job advert has $(C + M)(C + M - 1) / 2$ skill interactions. The emerging skills in the title string are ignored for simplification purposes.

Step 3: Skill interaction dataset generation
Given a set of job adverts, a skill interaction dataset with a set of skill interaction instances (advert ID, skill ID, skill ID) can be generated by using the skill interaction identification method.

Step 4: Conventional /emerging skill interaction network generation
Given a skill interaction dataset with the skill interaction instances, an undirected graph can be generated $G = (N, E)$, formed by the conventional skills and emerging skills $N = \{n_1, n_2, \ldots, n_N\}$ as vertices and a set $E = \{e_1, e_2, \ldots, e_E\}$ of edges $e_k = \{n_i, n_j\}$ when an advert contains the conventional skill i and emerging skill j. The weight of the edge W_{ij} is the number of job adverts in which the skill pair appears.

Step 5: Graphlet Decomposition
The generated skill interaction network can be decomposed into a 2-node and a 3-node graphlet (G0, G1 and G2), as shown in Figure 3. The G0 2-node graphlet has a skill pair (including Mobile Development - Cloud Computing, Mobile Development - Machine Learning, or Cloud Computing – Machine Learning) in an advert. Moreover, the G0 graphlets were labeled as $G0_L$ or $G0_S$ depending on if the advert title contained a conventional skill. The $G0_L$ 2-node graphlet has a skill pair with the mobile development skill in the advert title (including Mobile Development in Title – Cloud Computing or Mobile Development in Title – Machine Learning) and the mobile development skill node is highlighted in orange. The $G0_s$ 2-node graphlet has a skill pair with the mobile development in the advert context (including Mobile Development in Context – Cloud Computing or Mobile Development in Context – Machine Learning).

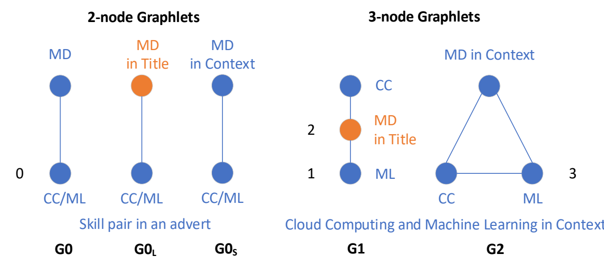


Figure 3: Graphlet Decomposition in 2-Node And 3-Node Graphlets (G0, G1 And G2). The 2-Node Graphlet G0 is also Labeled As $G0_L$ Or $G0_S$ Depending on if the Advert Title Contains a Conventional Skill. The Conventional Skill Which Appears in The Advert Title is Highlighted as Orange

In the 3-node graphlet G1, the advert title contained a conventional skill, and the advert context contained an emerging skill pair. The G1 3-node graphlet has three skills in an advert with the mobile development skill in the advert title and the cloud computing & machine learning skills in the advert context (Mobile Development in Title – Cloud Computing in Context – Machine Learning in Context). In the G1 3-node graphlet, the mobile development skill node is highlighted in orange. In the 3-node graphlet G2, the advert title did not contain a conventional skill, and the advert context contained the emerging skill and an emerging skill pair. The G2 3-node graphlet has three skills in the advert context (Mobile Development in Context – Cloud Computing in Context – Machine Learning in Context) (Espejo et al., 2020; Hocevar & Demsar, 2016).

Step 6: Role-based skill association strength measurement
The role-based skill association strength is measured for the conventional skill in leading and supporting roles based on the skill graphlets. Similar with the SCODIS scheme (Sibarani & Scerri, 2020), the co-occurrence frequency (Callon et al., 1991) was used to calculate the skill

association strength in job adverts, JA_{ij}, $JA_{ij} = (JC_{ij})^2 / (JC_i \times JC_j)$, where JC_{ij} was the number of job adverts containing the skill pair *i* and *j*; JC_i was the number of job adverts containing the skill *i*; JC_j was the number of job adverts containing the skill *j*.

The leading association strength LA_{ij} can be calculated based on $G0_L$ and G1 graphlets in which the conventional skill appears in the advert title, $LAij = (LC_{ij})^2 / (LC_i \times LC_j), 0 \le LA_{ij} \le 1$, where LC_{ij} is the number of $G0_L$ and G1 graphlets containing the skill pair *i* and *j*, LC_i is the number of $G0_L$ and G1 graphlets containing the skill *i*; LC_j is the number of $G0_L$ and G1 graphlets containing the skill *j*.

The supporting association strength SA_{ij} can be calculated based on extracted $G0_S$ and G2 graphlets in which the conventional skill only appears in the advert context, $SAij = (SC_{ij})^2 / (SC_i \times SC_j), 0 \le SA_{ij} \le 1$, where SC_{ij} is the number of $G0_S$ and G2 graphlets containing the skill pair *i* and *j*, SC_i is the number of $G0_S$ and G2 graphlets containing the skill *i*; SC_j is the number of $G0_S$ and G1 graphlets containing the skill *j*.

4.3 Online Course Content Analytics Module

In the online course content analytics module (Fig. 4), a python program was developed to build a pipeline for online course content dataset generation. The online course content was extracted from the Canvas learning management system modules pages. Through using the word frequency analysis on the online course content, the skill interactions between the conventional skill (extracted from the course title) and the emerging skills (extracted from the content) were identified. The identified conventional skill and emerging skill interactions were used to generate the skill interaction network. The skill association strength in course content CA_{ij} was then measured based on the generated skill interaction network, $CA_{ij} = (CC_{ij})^2 / (CC_i \times CC_j)$, where CC_{ij} was the number of skill interactions *i* and *j*; CC_i was the number of skill interactions containing the skill *i*; and CC_i was the number of job adverts containing the skill *j*. (Tamašauskaitė & Groth, 2022; Yu et al., 2021).

5. A CASE STUDENT IN AN ONLINE MOBILE APPLICATION DEVELOPMENT COURSE

The influence that mobile terminal devices have had on society and the economy has been, and will continue to be, transformative. Thus, mobile app development education plays a critical role in computing related curriculums (Aimicheva et al., 2020; Babb & Abdullat, 2012). Online mobile application development courses focus on the features and capabilities of the popular mobile platforms to develop a mobile application (Leidig et al., 2020). As industry continues to integrate emerging technologies such as cloud computing and machine learning into the mobile application development process, it is critical to have a better understanding of the dynamic skill demand in the industry and adaptively adjust course content to address the required skill set (Liu & Murphy, 2012).

In this case study, the proposed data-driven emerging skill network analytics framework was used to evaluate the market-oriented knowledge interaction in an online mobile application development course. The skill network analysis and visualizations were implemented to yield more insights for bridging the "Skill Gap" in program graduates.

In this pilot research project, the authors focused on the most in-demand skills associated with mobile development skill requirements. A set of queries with the keywords "developer mobile $emerging skill" were implemented on the Indeed.com website. The query results are shown as Table 1 in the Appendix. Through use of the network analysis and visualization tools *Pajek* (Batageli & Mrvar, 2022) and *VOSviewer* (van Eck & Waltman, 2022), a conventional/emerging skill network was generated based on the Indeed query results (Fig. 5). According to the co-occurrence of the conventional mobile development skill and various emerging skills in the job adverts (Fig. 6), the most in-demand emerging skills, cloud computing and machine learning, were selected as query keywords for data collection.

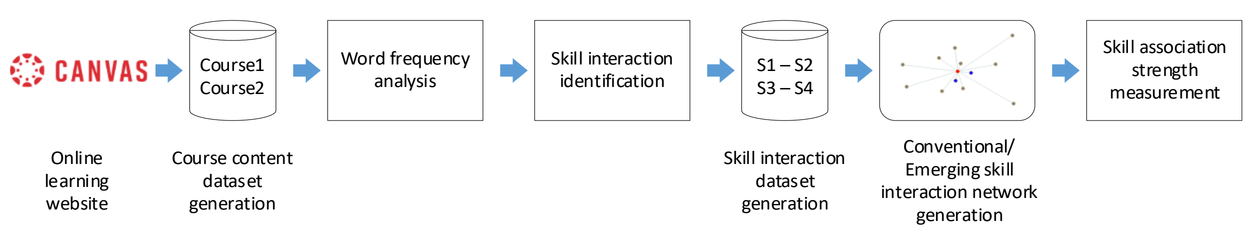


Figure 4: Online Course Content Analytics Module

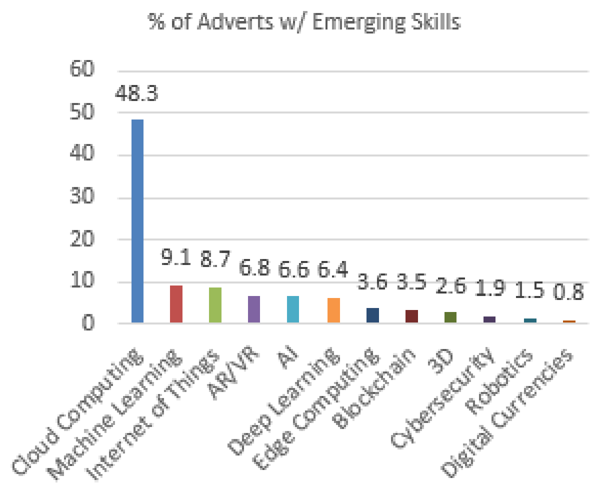


Figure 5: Percentage of Adverts with Various Emerging Skills in 33,067 Adverts Requesting Mobile Development Skills

5.1 Data collection

Job adverts derived from online job search/recruitment websites, such as Indeed, Monster, Glassdoor, FlexJobs, Ladders, AngelList, LinkedIn, Getwork, Scouted, Snagajob, etc. have been useful mining resources for identifying demand skills in the job market (Greenacre & Hastie, 2010; Wowczko, 2015; Zhang et al., 2017; Zhao et at., 2015). A python program with the emerging job skill network analytics module was implemented to collect online job advert data from the Indeed website. In this research, a set of html files containing 3,000 online job adverts were collected consisting of 1,000 adverts from the query "developer mobile cloud", 1,000 adverts from the query "developer mobile machine learning", and 1,000 adverts from the query "developer mobile cloud machine learning". The job advert skill interaction datasets, including the mobile-cloud dataset (MC dataset), mobile-machine learning dataset (ML dataset), mobile-cloud-machine learning dataset (MCL dataset), and AD dataset with all adverts, were then generated based on the collected html files.

Another python program containing the online course content analytics module was implemented to collect online course content data from the Canvas course website at the authors' institution. In this work, the html file of the Canvas Modules page listing the course content was collected for the online mobile application development course. The online learning content consisted of 20 learning modules, 22 hands-on projects, 10 individual/group assignments, 14 Zoom class sessions, and related learning materials/resources. The course content skill interaction datasets were created based on the Canvas html file.

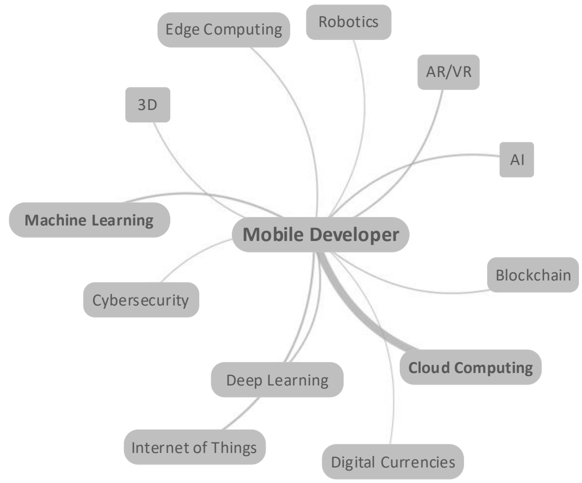


Figure 6: Indeed Query Results conventional/Emerging Skill Network

5.2 Modeling

Four skill networks with 13 skills as nodes and 78 weighted skill interactions as edges were built by using the job advert skill interaction datasets (MC, ML, MCL, and AD datasets) as depicted in Figure 7. The four skill networks have the same nodes and edges with different weights on the edges. However, the authors focused only on the skill interactions between the mobile development skills and associated most in-demand emerging skills (including cloud computing and machine learning). The number of skill interactions between the mobile development and cloud computing (M-C), mobile development and machine learning (M-L), and cloud computing and machine learning (C-L) various datasets are shown in Table 2.

	MC Dataset	ML Dataset	MCL Dataset	AD Dataset
M-C	392	20	274	686
M-L	2	434	278	714
C-L	0	32	123	155

Table 2: The Number of the Skill Interactions Between the Mobile Development and Cloud Computing (M-C), Mobile Development and Machine Learning (M-L), and Cloud Computing and Machine Learning (C-L) In the MC, ML, MCL, and AD Datasets

The 2-node and 3-node graphlets (G0, G1, and G2) which consist only of the mobile development, cloud computing, and machine learning skills, were then extracted from the generated skill networks, as shown in Table 3.

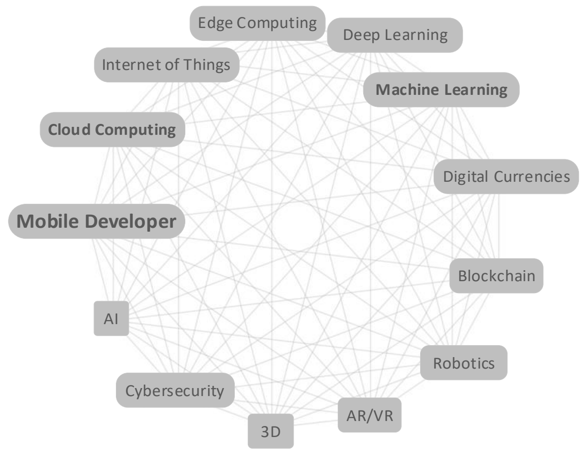


Figure 7: One of the Four Skill Networks with 13 Skills as Nodes and 78 Skill Interactions as Edges Built Using the Job Advert Skill Interaction Datasets (MC, ML, MCL, And AD). The Four Skill Networks Have the Same Nodes and Edges with Different Weights on the Edges.

Graphlet	MC Dataset	ML Dataset	MCL Dataset	AD Dataset
G0	394	454	552	1400
G1	0	0	0	0
G2	0	32	123	155

Table 3: The Number of the 2-Node and 3-Node Graphlets (G0, G1, and G2) Consisting Only of the Mobile Development, Cloud Computing, and Machine Learning Skills from Various Skill Networks

In this pilot research project, we extracted the top 8 high-frequency words from the Canvas Modules Page of the online course (41,520 total words of which 386 were unique). A skill network with 9 skills as nodes and 8 weighted skill interactions as edges was created by using the online course content analytics module, as shown in Figure 8. The weighted edges are represented as the top 8 high-frequency skills and the conventional skill "Mobile Development" in the Canvas Course Modules html file of the online mobile application development course (Fig. 9).

6. ANALYSIS OF NETWORKS

A set of analyses were developed based on four generated skill networks derived from various online advert datasets including MC, ML, MCL, and an Integrated AD dataset. As shown in Figure 10, the cloud computing-related mobile developer job adverts in the MC dataset had:

- higher demand of individual Mobile-Cloud Interaction computing skills (39% adverts),
- very low demand of individual Mobile-Machine Learning interaction skills (0.2% adverts), and
- no demand for Cloud Computing-Machine Learning interaction skills (no adverts).

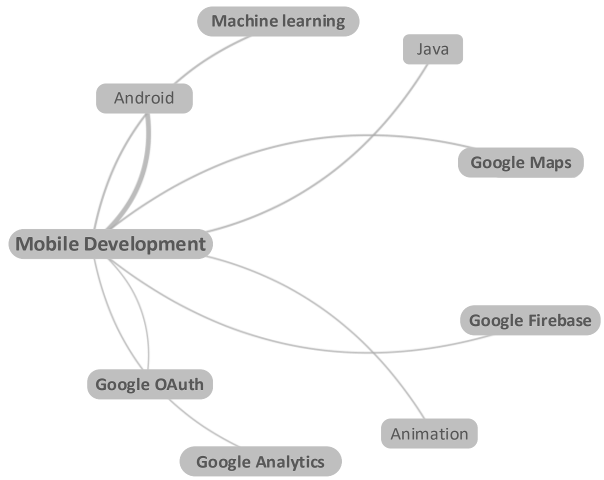


Figure 8: A Skill Network with 9 Skills as Nodes and 8 Weighted Skill Interactions as Edges.

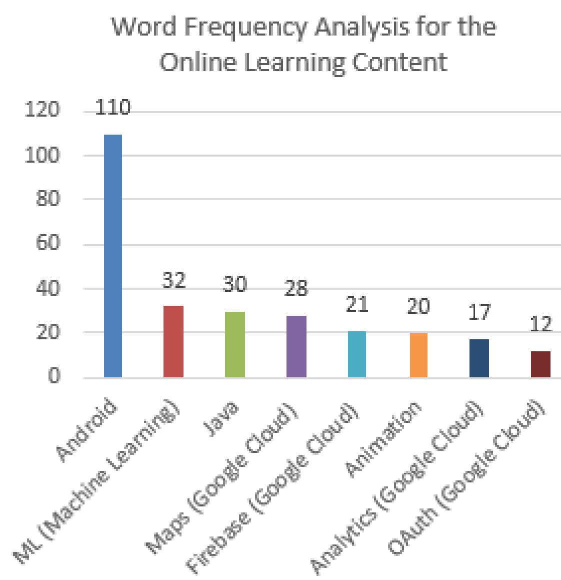


Figure 9: Top 8 High-Frequency Words Extracted from the Online Mobile Application Development Canvas Course (Total 41,520 Words of which 386 Were Unique)

The machine learning-related mobile developer job adverts in the ML dataset had:

- very low demand of individual Mobile-Cloud Interaction computing skill (2% adverts),
- higher demand of individual Mobile-Machine

Learning interaction skills (43.4% adverts), and
- very low demand for Cloud Computing-Machine Learning interaction skills (3.2% adverts).

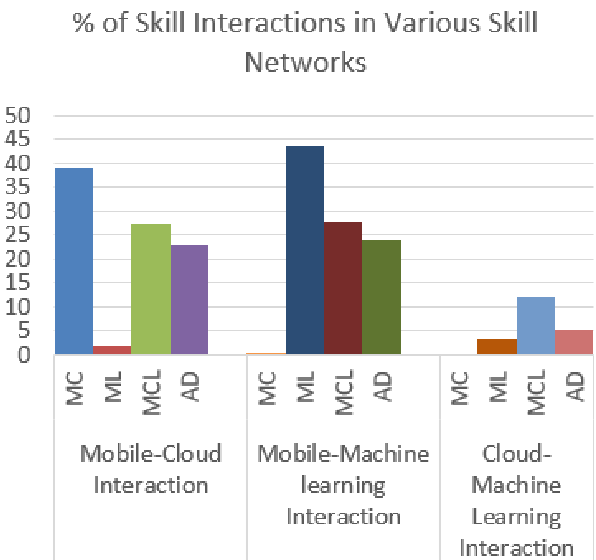


Figure 10: Percentage of Different Skill Interactions in Various Skill Networks (MC, ML, MCL, and AD)

The cloud computing- and machine learning-related mobile developer job adverts in the MCL dataset had:

- moderate demand of individual Mobile-Cloud Interaction computing skill (27.4% adverts),
- moderate demand of Mobile-Machine Learning Interaction skills (27.8% adverts), and
- moderate demand for Cloud Computing-Machine Learning Interaction skills (12.3% adverts).

According to the generated skill networks derived from online job adverts, the job market also requests the combined skills of cloud computing and machine learning for mobile-related developers.

A set of 2- and 3-node graphlets were generated from four skill networks (MC, ML, MCL, and AD). As depicted in Figure 11:

- cloud computing-related mobile developer job adverts had a higher demand for individual cloud computing skills (having 2-node graphlet (G0) in 39.4% adverts);
- machine learning-related mobile developer job adverts had a higher demand for individual machine learning skills (having 2-node graphlet (G0) in 45.4%); and
- cloud computing- and machine learning-related mobile developer job adverts had a:
  - higher demand of individual cloud computing or machine learning skills (having 2-node graphlet (G0) in 55.2% adverts), and
  - moderate demand of cloud computing-machine learning combined skills (having 3-node graphlet (G2) in 12.3% adverts).

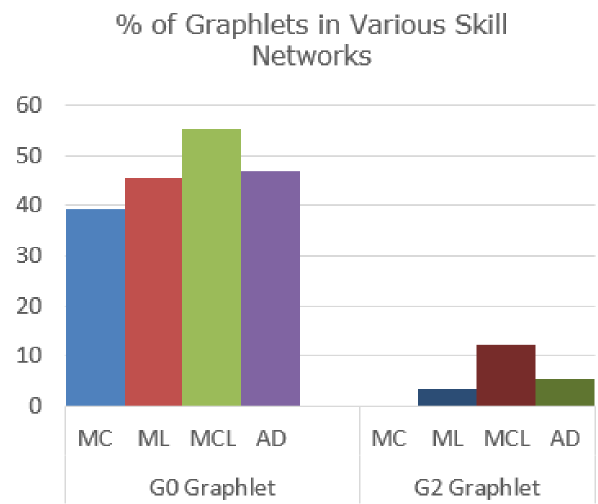


Figure 11: Percentage of 2-Node Graphlet (G0) and 3-Node Graphlet (G2) in Various Skill Networks (MC, ML, MCL, and AD)

Through identifying the skill interactions between the job title and context in the job adverts, the roles of the skills were recognized in the generated skill networks. The following results were brought to light through the analysis:

- If the job title contained "mobile" as a keyword, the job advert was a mobile development-centric job advert. The mobile development skill was a leading skill in this job advert, the cloud computing and/or machine learning skills in the context were supporting skills.
- If the job context contained "mobile" as a keyword instead of the job title, the job advert was a mobile development-related job advert. The mobile development skill was also a supporting skill in this job advert.
- As shown in Figure 12, the most collected job adverts were mobile development-related with the conventional mobile development skill as a supporting skill (86.8% in total adverts). Moderate cloud-related developer job adverts required mobile development skills as a leading skill (27.1% in the MC skill

network).

- Most machine learning-related developers adverts required mobile development skills as a supporting skill (93.2% in the ML skill network and 94.2% in the MCL skill network).
- As depicted in Figure 12 and Figure 13, the mobile development-related job adverts had a higher demand for the combination of individual machine learning skills and cloud computing-machine learning skills.

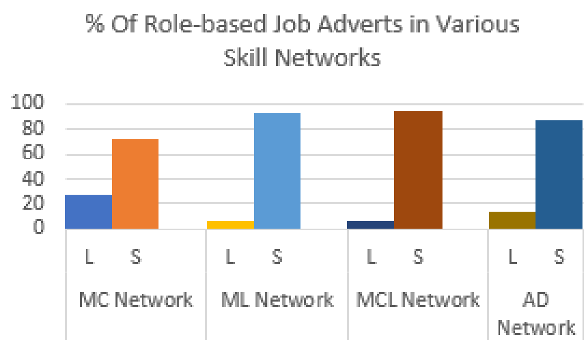


Figure 12: Percentage of Mobile Development-Centric Job Adverts (L) and Mobile Development-Related Job Adverts (S) in Various Skill Networks (MC, ML, MCL, and AD)

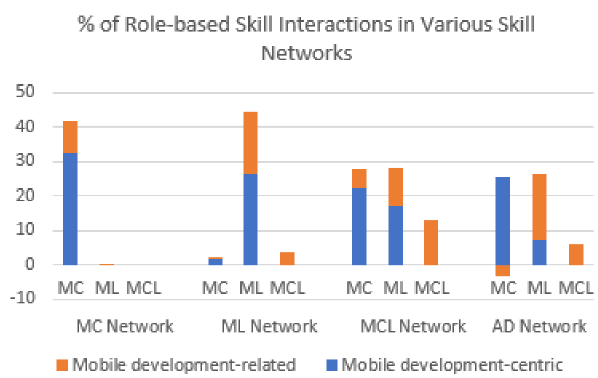


Figure 13: Percentage of Role-Based Skill Interactions in Various Skill Networks

The next section describes the application of the network to the skill content taught in a mobile development course at the authors' institution.

7. APPLICATION OF THE NETWORK TO A MOBILE DEVELOPMENT COURSE

The skill network generated from the course online content consisted of eight high-frequency words associated with the conventional mobile application development skills. Five of the words were derived from the services provided by the Google Cloud Platform (GCP) including Google *Maps*, Google *Firebase*, Google *Analytics*, and Google *OAuth*. The interaction strength between the mobile application development skill and the cloud computing skill was calculated as a sum of the frequency of the five words (Maps, Firebase, Analytics, and OAuth). The course content skill network was then generated consisting of 6 nodes and 5 weighted edges. The skill interaction strength is shown in Figure 14.

The skill association strength was measured through multiple skill networks including the:

- course content skill network,
- AD skill network containing all collected job adverts,
- leading skill network with all mobile-centric adverts, and
- supporting skills network with all mobile-related adverts.

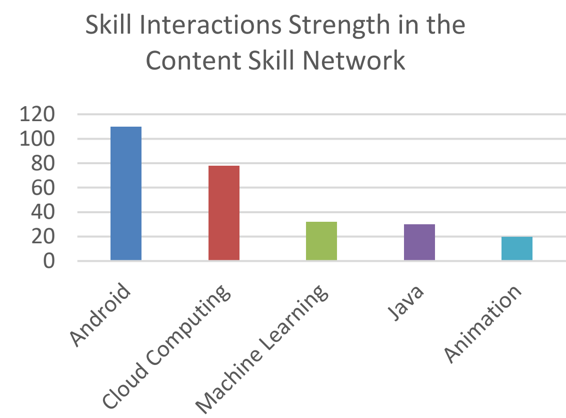


Figure 14: Skill Interactions Strength in the Course Content Skill Network with 6 Nodes (Including Mobile Development, Cloud Computing, Machine Learning, Java, and Animation) and 5 Weighted Edges

As shown in Figure 15, the skill association strength of the conventional mobile application development skills and the emerging cloud computing skills (0.29 in MC interactions) were higher than the strength of the mobile application development skills and the emerging machine learning skills (0.12 in ML interactions) in the course content skills network.

In the next sections, the authors describe the changes that were made to the course based upon the results of the analysis.

8. OUTCOMES

Based on the market-oriented skill network analysis and the course content-based skill network analysis, more cloud computing- and machine learning-related learning content should be integrated into the author's current online mobile application development course. In Spring 2022, the online mobile development course was redesigned and enhanced with more emerging skill-related learning content. As shown in Table 4 in the Appendix, in addition to the original 10 mobile development hands-on projects from the previous semester, an additional 8 hands-on mobile development projects with emerging skills were incorporated including cloud computing (4), machine learning (2), cybersecurity (1), and IoT (1). Another 2 hands-on mobile development projects with data analytics skills and animation skills were also integrated. The comprehensive learning content of the new additional hands-on projects was provided on the Canvas course website including lecture notes, recorded lecture videos, study guides, project manuals, recorded project instruction videos, supplemental materials, external resource links, and forums. In future course content analysis and application of the framework, data will be extracted from these resources as well.

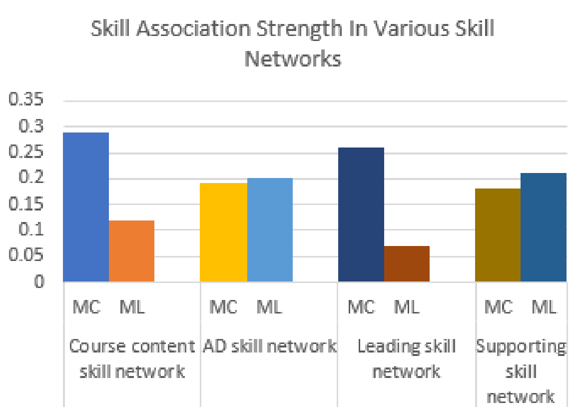


Figure 15: Skill Association Strength in Various Skill Networks Including: Course Content Skill Network, AD Skill Network w/ All Collected Job Adverts, Leading Skill Network w/ All Mobile-Centric Adverts, and Supporting Skill Network w/ All Mobile-Related Adverts

9. CONCLUSION

Ensuring that faculty teach the requisite emergent technical skills that employers need graduates to have as they enter the marketplace is imperative. In this paper, the authors described the development of a data-driven analytics framework that can be used for evaluating emerging skill clusters in online job adverts. The framework was then applied to the content of a mobile application development course taking into account current online job advert skill set requirements.

The focus of the authors in this pilot research project was to provide a data-driven framework, and technical solution, to provide some high-level graphics and insights based on dynamic job market requirements for IS/CS educators. However, adjustments to teaching materials are contingent on instructors' understanding of the materials, instructional strategy, and teaching philosophy. More evaluative work needs to be done in the future regarding the impact of job market requirements in the IT industry in IS/CS program curriculum design.

In future research, the authors would like to apply the framework to further analyze other IS/IT curriculum as well as incorporate audio data extracted from course lecture videos. In addition, the model can be extended to conduct a data-driven market-oriented skill valuation assessment across higher education modalities (conventional/online/hybrid coexistence) during the post pandemic digital transformation. The authors would also like to introduce education costs/tuition fees as a variable into the framework and compare the results with the market-oriented skill valuation framework.

10. REFERENCES

Agarwal, N., & Ahmed, F. (2017). Developing collective learning extension for rapidly evolving information system courses. *Education and Information Technologies, 22*(1), 7-37.

Aimicheva, G., Zh, K., Zh, O., Tokzhigitova, N., & Akimova, S. (2020). A spiral model teaching mobile application development in terms of the continuity principle in school and university education. *Education and Information Technologies, 25*(3), 1875-1889.

Alenezi M. (2021). Deep dive into digital transformation in Higher Education Institutions. *Education Sciences, 11*(12), 770.

Babb, J. S., & Abdullat, A. (2012). The need for mobile application development in IS curricula: An innovation and disruptive technologies perspective, *Information Systems Education Journal, 10*(1), 61-74.

Batagelj, V., & Mrvar, A. (2022). Pajek: Program for Large Network Analysis (Version 5.15) [Computer Software] Ljubljana, Slovenia

Available from http://mrvar.fdv.uni-lj.si/pajek/

Callon, M., Courtial, J.P., & Laville, F. (1991). Co-word analysis as a tool for describing the network of interactions between basic and technological research: The case of polymer chemistry, *Scientometrics*, *22*(1), 155-205.

Clear, A., & Parrish, A. (2020). Computing Curricula Report CC2020. Global Guidelines for Baccalaureate Degrees in Computing. New York, NY: ACM and IEEE.

Cowan, R. & Jonard, N. (2001). Knowledge creation, knowledge diffusion and network structure. In Kirman, A. Zimmermann, JB. (eds). Economics with Heterogeneous Interacting Agents. Lecture Notes in Economics and Mathematical Systems 503. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-56472-7_20

Cummings, J., & Janicki, T. N. (2020). What skills do students need? A multi-year study of IT/IS knowledge and skills in demand by employers. *Journal of Information Systems Education, 31*(3), 208-217.

Dadzie, A. S., Sibarani, E. M., Novalija, I., & Scerri, S. (2018). Structuring visual exploratory analysis of skill demand. *Journal of Web Semantics*, 49, 51-70. https://doi.org/10.1016/j.websem.2017.12.004 .

de Blas, C. S., Gonzalez, D. G., & Herrero, R. C. (2021). Network analysis: An indispensable tool for curricula design. A real case-study of the degree on mathematics at the URJC in Spain, *PLoS One, 16*(3), 1-21. DOI: 10.1371/journal.pone.0248208.

da Silveira Bueno, C. Masago, F. K., da Silveira, J. M (2021). Bioenergy emerging paradigm: Collaborative networks and determinants in the upgrading process of technological frontiers. *African Journal of Science Technology Innovation and Development*. https://doi.org/10.1080/20421338.2021.1960540

Espejo, R., Mestre, G., Postigo, F., Lumbreras, S., Ramos, A., Huang, T., & Bompard, E. (2020). Exploiting graphlet decomposition to explain the structure of complex networks: The GHuST framework. *Scientific Reports*, *10*(1), 12884. https://doi.org/10.1038/s41598-020-69795-1

Greenacre, M., & T. Hastie. (2010). Dynamic visualization of statistical learning in the context of high-dimensional textual data. *Web semantics: Science, services and agents on the World Wide Web, 8*(2-3), 163-168. 10.1016/j.websem.2010.03.007

Hocevar, T., & Demsar, J. (2016). Computation of graphlet orbits for nodes and edges in sparse graphs. *Journal of Statistical Software*, *71*(10), 1-24. https://doi.org/10.18637/jss.v071.i10

Koong, K. S., Liu, L. C., & Liu, X. (2002). A study of the demand for information technology professionals in selected internet job portals. *Journal of Information Systems Education, 13* (1), 21-28.

Leidig, P. M., Salmela, H., Anderson, G., Babb, J., Gardner, L., Nunamaker, J. F., Scholtz, B., Shankararaman, V., Sooriamurthi, R., Thouin, M., & de Villiers, C. (2020). IS 2020: Developing an ACM/AIS information systems model curriculum for undergraduate programs. *In International Conference on Information Systems, ICIS 2020 - Making Digital Inclusive: Blending the Local and the Global)*. Association for Information Systems.

Liu, X. M., & Murphy, D. (2012). Tackling an IS educator's dilemma: A holistic model for "When" and "How" to incorporate new technology courses into the IS/IT curriculum. *SAIS 2012 Proceedings*. 31. https://aisel.aisnet.org/sais2012/31

Mitchel, A. (2022). Addressing the dynamic information technology landscape: A course reflection. *Communications of the Association for Information Systems,* 50, pp-pp. https://doi.org/10.17705/1CAIS.05020

Morris, G. J., Fustos, J. T., & Haga, W. (2018). Connecting the dots and nodes: A survey of skills requested by employers for network administrators. *Information Systems Education Journal, 16*(1), 4-12.

Sibarani, E. M., & Scerri, S. (2019). Generating an evolving skills network from job adverts for high-demand skillset discovery, In: Cheng, R., Mamoulis, N., Sun, Y., Huang, X. (eds) *Web Information Systems Engineering – WISE 2019. WISE 2020. Lecture Notes in Computer Science*, 11881. Springer, Cham. https://doi.org/10.1007/978-3-030-34223-4_28

Sibarani, E. M., & Scerri, S. (2020). SCODIS: Job advert-derived time series for high-demand skillset discovery and prediction. In: Hartmann, S., Kung, J., Kotsis, G. Tjoa, A. M., Khalil, I. (eds) *Database and Expert Systems*

Applications. DEXA 2020 Lecture Notes in Computer Science, 12392. Spring, Cham. https://doi.org/10.1007/978-3-030-59051-2_25

Smith, D., & Azad, A. (2014). Analyzing computer programming job trend using web data mining. *Issues in Informing Science and Information Technology*, 11, 203-214 Retrieved June 8, 2022, from http://iisit.org/Vol11/IISITv11p203-214Smith0494.pdf

Sun, Y., Zhuang, F., Zhu, H., Zhang, Q., He, Q., & Xiong, H. (2021). Market-oriented job skill valuation with cooperative composition neural network. *Nature Communications, 12*(1), 1992, https://doi.org/10.1038/s41467-021-22215-y

Tamašauskaitė, G. & Groth, P. (2022). Defining a knowledge graph development process through a systematic review. *ACM Transactions on Software Engineering and Methodology*. *1*(1) 40 pages. https://doi.org/10.1145/3522586

van Eck, N. J., & Waltman, L. (2022). VOSviewer (Version 1.6.17) [Computer software]. Leiden, Netherlands. Available from https://www.vosviewer.com/

Wijesinghi, C. (2022). How knowledge diffusion in university-industry collaboration can foster ICT innovations in Sri Lanka, (Publication No. 978-91-7911-836-5) [Doctoral dissertation, Stockholm University] Retrieved June 9, 2022, from https://www.diva-portal.org/smash/get/diva2:1647417/FULLTEXT01.pdf

Wowczko, I. A. (2015). Skills and vacancy analysis with data mining techniques. *Informatics*, *2*(4), 31-49. https://doi.org/10.3390/informatics2040031

Yu, X., Stahr, M., Chen, H. & Yan, R. (2021). Design and implementation of curriculum system based on knowledge graph. *IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 767-770, doi: 10.1109/ICCECE51280.2021.9342370.

Zhao, M., Javed, F., Jacob, F., & McNair, M. (2015). SKILL: A System for skill identification and normalization, In *Innovative Applications of Artificial Intelligence*, *29*(2), 4012-4017. https://doi.org/10.1609/aaai.v29i2.19064

Zhang, L., Thalhammer, A., Rettinger, A., Färber, M., Mogadala, A., & Denaux, R. (2017). The xLiMe system: Cross-lingual and cross-modal semantic annotation, search and recommendation over live-TV, news and social media streams, *Journal of Web Semantics*, *46–47,* 20-30. https://doi.org/10.1016/j.websem.2017.03.002 .

Zollman, K. J. S. (2013). Network epistemology: Communication in epistemic communities. Philosophy Compass. *8*(1), 15-27.

APPENDIX - TABLES

ID	Role	Conventional skill	Emerging skill	Query keyword	Co-occurrence
1	Developer	Mobile	Cloud Computing	developer mobile cloud	15,982
2	Developer	Mobile	Machine Learning	developer mobile machine learning	3,020
3	Developer	Mobile	Internet of Things	developer mobile iot	2,879
4	Developer	Mobile	AR/VR	developer mobile ar vr	2,236
5	Developer	Mobile	AI	developer mobile ai	2,186
6	Developer	Mobile	Deep Learning	developer mobile deep learning	2,112
7	Developer	Mobile	Edge Computing	developer mobile edge	1,197
8	Developer	Mobile	Blockchain	developer mobile blockchain	1,156
9	Developer	Mobile	3D	developer mobile 3d	873
10	Developer	Mobile	Cybersecurity	developer mobile cybersecurity	629
11	Developer	Mobile	Robotics	developer mobile robotics	488
12	Developer	Mobile	Digital Currencies	developer mobile currency	281
13	Developer	Mobile	N/A	developer mobile	33,067

Table 1: A Set of Queries with the Keywords "Developer Mobile $Emerging Skill" On Indeed Website.

ID	Module	Conventional skill	Emerging skill	Hands-on project
1	0	Android		Building work environment project
2	1	Java		Basic Java programming project 1
3	2	Java		Basic Java programming project 2
4	3	Android		Test run project
5	4	Android		Android Studio welcome project
6	4	Android		MPAndroidChart project
7	5	Android		TipCalculator project
8	5	Android	Cloud	Google Charts project
9	6	Android		FlagQuiz project
10	6	Android		Android View Animation project
11	7	Android		Doodlz project
12	7	Android	Cloud	Google Maps project
13	8	Android		CannonGame project
14	8	Android	Cloud	Google OAuth login project
15	9	Android		WeatherViewer project
16	9	Android	Machine Learning	Text and facial features recognition with Google ML Kit
17	10	Android		Twitter Search project
18	10	Android	Machine Learning	Text translation with Google ML Kit project
19	11	Android		AddressBook project
20	11	Android	Cloud	Google analytics with Firebase project
21	12	Android	IoT	Flutter Android application project
22	12	Android	Cybersecurity	Rooted Android Studio AVD project

Table 4: The 22 Hands-On Projects in the Online Mobile Application Development Course in Spring 2022. The New Added Hands-On Projects Were Highlighte

Examining the Number of Concepts Students Apply in the Exam Solutions of an Introductory Programming Course

Pratibha Menon
menon@pennwest.edu
Department of Computer Science and Information Systems
Pennsylvania Western University, California
California, PA, 15419, USA

Abstract

Instruction in an introductory programming course is typically designed to introduce new concepts and to review and integrate the more recent concepts with what was previously learned in the course. Therefore, most exam questions in an introductory programming course require students to write lines of code that contain syntactic elements corresponding to the programming concepts covered during the instruction. This study investigates the number of concepts involved in the exam problems of an introductory Java programming course. In addition, this study compares how the increase in the number of concepts correlates with the ability of students to write error-free lines of code. The instructional method adopted in this study focuses on providing students with a problem-solving schema and a resultant programming plan that integrates many concepts to meet the problem's goal. Results from this study indicate that as the course progresses through the semester, students, on average, apply appropriate problem-solving schemas and programming plans to produce more error-free lines of code, despite an increase in the concept count in the problems. Furthermore, the exam problems later in the course repeat the application of cluster concepts that have appeared in the past exam. This paper illustrates how programming is a cumulative skill and that repeating and building upon the applications of these concept clusters several times through the course increases the likelihood that students will produce more correct lines of code as the semester progresses.

Keywords: Concepts, Introductory-Programming, Lines-of-code, Exams, Program.

1. INTRODUCTION

Failure rates in introductory programming courses have prompted several researchers to identify the causes that make these courses difficult for students (Watson & Li, 2014; Medeiros et al., 2019; Bennedsen & Caspersen, 2019). One thread of research has explored the types of assessment used in introductory programming courses to determine the factors that make the test items difficult (Zur & Vilner, 2014; Ford & Venema, 2010). Exams in a programming course typically consist of tasks designed to assess how well students can apply various programming constructs to solve problems. The difficulty of a programming task in an exam item may be evaluated subjectively from student self-assessments or objectively by collecting data about the problems' characteristics and their solutions (Braarud, 2001).

Prior studies have found that the difficulty of exam questions in introductory programming courses taught by different instructors depended on various measures of complexity such as the degree of explicitness, reference to an external and unfamiliar domain, hard-to-learn concepts, linguistic complexity, intellectual complexity level based on Bloom's taxonomy. (Sheard et al., 2011; Sheard et al., 2013; Harland, D'Souza & Hamilton, 2013). In these studies, the complexity measures were evaluated based on the instructors' perceptions, and the exam questions' resulting difficulty was inferred through the students' marks.

Studies also noted that exam questions in introductory courses were predominantly composed of integrative code-writing questions that required students to apply multiple concepts (Petersen et al., 2011). Evaluating questions' content and cognitive requirements indicate that students must internalize a large amount of introductory programming content and gain enough practice solving problems to succeed in the exams. While academics can evaluate the difficulty of the exams, they tend to underestimate the total number of concepts used versus those evaluated (Simon et al., 2012). For example, a question on loops requires students to master basic concepts such as Boolean logic, variables, data types, operators, and the associated syntax before constructing a loop that solves a given problem. Some programming concepts are so fundamental that they are used in every code-writing instance and must be committed to long-term memory.

Cognitive load theory explains that concepts not fully internalized must be reasoned in the working memory (Ericsson & Kintsch, 1995; Berssanette & de Francisco, 2022). In addition, the mind is limited in its ability to work with multiple concepts simultaneously, and therefore, students who need more mastery of fundamental concepts face an increasing cognitive burden (Muller et al., 2007). Results from prior studies argue that we may be asking students in introductory programming courses to master too many concepts in a short time (Goldman et al., 2010).

This study investigates the intrinsic cognitive load of course contents in an introductory programming course by quantifying the conceptual complexity of the exam problems used to assess student learning. The conceptual complexity of an exam problem is measured by the number of distinct concepts contained in an optimal code solution. Instruction in an introductory programming course takes place by introducing students to new concepts and integrating them with the previously taught concepts. Therefore, exam questions are formulated to assess the conceptual knowledge gained by students by evaluating how well students learn to integrate and apply these concepts to solve problems.

This study investigates how the cognitive load introduced in learning programming concepts impacts the ability of students to produce correct code. First, the conceptual complexity of the course contents is measured by counting the concepts expected to be applied in each of the solutions to the exam problems. Then, the learning outcome is measured as the correctness of the lines of code of the solutions produced by students at three different points through a fifteen-week semester.

The approach of evaluating the conceptual complexity of exam problems by identifying and counting the number of concepts applied in the expected solution, as explained in this paper, could be used by instructors to objectively gauge the difficulty of exam questions in their introductory programming courses. This paper also examines the possibility of using lines of code as a reasonably good metric within an introductory programming course to score the exam solutions for a student's ability to write code by applying the required concepts. The research question of this study is formulated as follows:

RQ1: How do the number of distinct concepts students apply to solve exam problems increase as the semester progresses in an introductory programming course?

RQ2: How do the number of concepts students apply to solve a programming problem correlate with their ability to produce an error-free solution as they obtain instruction and practice to solve application-based problems in an introductory programming course?

This study takes place in a live classroom with class lectures and a detailed code walkthrough demonstrating the instructor's practices in applying the concepts to solve problems. The introductory programming course investigated in this study has three monthly exams that test the ability of students to recall, analyze and apply their conceptual knowledge to write code sequences. The findings of this study have implications for designing instruction that supports instructors and students in taming the complexity of integrating many concepts in programming solutions.

2. THE COURSE CONCEPTS AND LINES OF CODE

Course exams are a valuable proxy for deriving curricular expectations and determining what instructors understand as essential. This study explores the concepts used in the exam questions of an introductory Java programming course. While no standard concept inventory exists for introductory programming courses, researchers have used varying methods to derive a list of essential concepts. For example, Tew and Guzdial have used the contents of textbooks to identify ten critical topics (Tew & Guzdial, 2010). Schulte and Bennedsen shortlisted 28 topics from the literature and asked instructors to rate the difficulty and relevance of each (Schulte & Bennedsen, 2006). A prior study by Petersen et al. evaluated exam contents and observed that the number of concepts considered by instructors while evaluating and grading programming solutions in an exam is fewer than those used to construct the program (Petersen et al., 2011).

This study draws the concepts from a concept inventory created for CS1 courses (Goldman et al., 2010), as depicted in Table 1.

Concept	
Program Structure	Arithmetic Operators
Method Structure	Assignment Operator
Method Parameter	Operator Precedence
Method Return	Proper use of parenthesis
Method Call	Expression
Syntax	Statement
Data Types	Conditionals
Variables	Decision Structures
Literals	Loops
Boolean Operators	Nested Loops
Variable Scope	String methods

Table 1. List of Concepts

All the concepts displayed in Table 1 are treated as being equally difficult, although studies have qualitatively identified certain threshold concepts that are more critical than others in the learning process (Meyer & Land, 2005; Sanders & McCartney, 2016). However, the results of these studies could only broadly identify the threshold concepts, for example, as pointers or object-oriented programming. Both these topics were not included in the curriculum of the introductory course, whose content is investigated in this paper. A study by Cherenkova et al. (2014) investigated a large dataset of CS1 code-writing attempts and found that certain straightforward application of concepts tend to be problematic even towards the end of the term.

Thus far in computing education, evaluating code complexity of solutions has either involved expert evaluation or the use of convenient metrics, such as the number of syntactic elements in a piece of code. While metrics-driven software engineering has fallen out of favor, they are a convenient quantitative method for measuring code (DeFranco & Voas, 2022). One popular metric, lines of code, is commonly used to measure developers' productivity. Lines of code is also an intuitive metric for measuring software size since its effect can be visualized. For example, lines of code could be used to count a program's volume of instructions (or statements). However, not all lines of code in a Java program may terminate in a semi-colon. For example, a for-loop does not contain a semi-colon but forms a line of code containing an executable entity.

Lines of code may be composed differently by novice and professional programmers (Kramer et al., 2017). Skilled developers can apply more syntactic entities with far less code. However, most novice developers, such as the students who attend an introductory programming course, only pack a physical line of code with a few logical constructs. While learning, it is easier for novices to comprehend and write code if each physical line contains the application of fewer logical constructs and the program is written in a step-by-step manner, as a logical sequence, using separate lines.

This study considers two types of heuristics - the number of concepts expected to be applied in an optimal solution and the expected lines of code to study how these metrics could infer the problem's complexity. The lines of code count all the instances of using syntactic elements that correspond to the concepts used to solve the problem. On the other hand, the concept count only considers the "distinct concepts" used to formulate an optimal solution. It is important to note that the number of distinct concepts applied in the solution should be optimal, which means these concepts are the ones whose application is necessary and cannot be avoided in the solution. This study explores how these two heuristics - the number of distinct concepts expected in an optimal solution and the total number of lines of code in the students' solution, could gauge students' learning progress to solve increasingly more significant problems in a course during a semester.

3. THE STUDY

This study takes place in a 15-week introductory Java Programming class in an undergraduate-level Computer Information Systems program at a public university. The course has three-unit exams that are spread out throughout the course. Exam1 covers the topics of decision structures. Exam 2 focuses on loops, and Exam 3 tests the ability of students to modularize their code using methods.

The Exams

The exams comprised coding problems that required students to apply their conceptual knowledge. Some questions only require students to analyze code. Most questions, however, require students to write a code solution. The exam questions were of variable points, and scores were assigned to each question based on the correctness of the code lines expected in the solution. In addition, students are given partial points to a solution based on the percentage of the number of lines of code answered correctly, compared to the lines of code expected in a correct solution. Given the stringent time allotted to complete the exams, no open-ended questions could have resulted in a high degree of code variability in the solutions. Each hour-long, closed-book exam was conducted in a classroom, and the exam was strictly timed and proctored. Students access and submit their exams through the course learning management system. Furthermore, due to the time limits of the exams, students were not asked to use a compiler to run their solutions during the exam. The exam's primary intent was to test students' ability to recall the syntax and apply their conceptual knowledge to write java program statements.

Points carried by an exam question correlated with the number of lines of code students had to write or analyze. For example, short answer questions required students to write or analyze one or two lines of code. Medium-sized questions had solutions that contained between 5-11 lines of code. A more extensive solution had about 12-26 lines of code. A summary of the characteristics of the exam questions for the three exams is given in Table 2. The upcoming sections of this paper will illustrate how the conceptual complexity of the exam questions evolves between the three exams. It must be emphasized that the exam questions were created such that the program solutions resembled a multi-step problem solving process, where each step involves application of a different cluster of concepts. Therefore, care was taken to ensure that larger code sizes in the exam solutions did not just result from repetition of similar statements involving the same group of concepts.

	Exam1	**Exam2**	**Exam3**
Duration	1 hour	1 hour	1 hour
Max points	50	70	100
# of questions	8	8	5
Points/ question	between 5 and 20	between 5 and 20	between 10 and 50
Approx # of expected Lines of Code / question	between 1 and 16	between 1 and 16	between 7 and 25

Table 2 – Exam summary

The Instruction

Before each exam, students were exposed to the exam topics via class lectures, code demonstrations, and weekly assignment exercises. The course contents are covered in four modules. Appendix D shows the assignment problems from each module along with the key concepts covered in that module. Through these learning activities, students are exposed to various problems that apply the concepts listed in Table 1. Appendix D also categorizes problem into various types such as calculators, checkers, counters etc. The code demonstrations used to instruct problem solving methods in class, covered several application scenarios and code development techniques. Every code walkthrough thoroughly explained a programming problem and solutions using a program plan that reflected the instructor’s problem-solving schema. Appendix C also shows how a simple problem could be broken down into various steps to develop a code walkthrough. Appendix C also shows a flow chart used by the instructor to plan the code walk through for any given problem.

The assignment problems provided means for students to solidify their conceptual understanding and apply (or modify) their problem-solving schemas to solve similar problems from a different context. Appendix D shows a sequence of assignment problems that also allows the reuse the concepts covered in the previous assignments. The assignment problems are similar in scope and scale to the ones whose solutions are explained in the code demonstrations. The exams help the instructor evaluate how correctly students transfer the problem-solving schemas and program plans involving multiple concepts to fit the specific context of an exam problem.

Exam solutions of 25 students who attended all three exams were collected. Any information identifying a student was removed from the solutions. Student submissions were not matched across the exams. Students' answers were scored for correctness by comparing them with the expected statements and syntax of the lines of code in the instructor's solutions. Every line of code that formed a statement was checked for correctness and assigned a point only if there were no errors.

4. RESULTS

Concepts in the Exam Questions

An analysis of the exam questions by the course instructor revealed all the concepts that a student needs to apply to solve each exam question. Figure 1 shows that the total number of concepts increased in the later exams. Figure 1 names each exam problem using the exam number and the problem number. For example, E3P5 stands for Exam 3, problem 5. Appendix A shows the mapping of each exam problem to the distinct concepts that need to be applied to write an optimal solution. Appendix B lists a partial list of questions from the three exams.

Figure 1 also shows the approximate number of lines of code students were expected to write or analyze in each exam problem. It can be observed from Figure 1 that even a single line of code could contain syntactic elements that represented multiple concepts. For example, problem E1P1 (described in Appendix B) required students to analyze a statement that contained a compound Boolean expression containing comparison and logical operators. While students were evaluated based on their understanding of Boolean expressions, they also needed to understand several foundational concepts, such as operator precedence, proper use of parenthesis, and the Java syntax used to comprehend a Boolean expression.

In Exam 1, problems 1, 2, and 3 (listed as E1P1, E1P2, and E1P3) required students to analyze a given statement, and problems 4, 5, 6, 7, and 8 (depicted as E1P4 – E1P8) required students to write lines of code using the concepts required to write if-else or switch statements.
Appendix B describes some of the questions from Exam 1. For example, writing lines of code that contain decision structures and the actions that follow the truth value of each conditional expression in the decision structure brings together 10 – 14 concepts, as observed in the concept mapping table in Appendix A. The bars corresponding to E1P5, E1P6, E1P7, and E1P8 in Figure 1 also show the many concepts used to solve these problems.

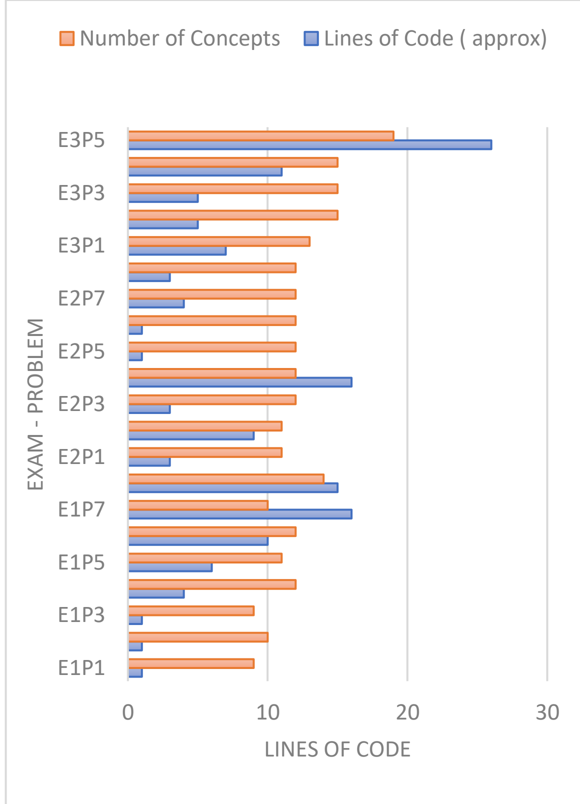


Figure 1. Concept count and the expected lines of code for exam questions.

Exam 2 required students to know how to write applications that use while, do-while, and for loops. Programs that included loops also contained foundational concepts such as variables, Boolean expressions, different types of operators, and simple conditional statements. To apply loops in a program, students also need knowledge of data types and syntax rules to compose expressions and statements. The number of lines of code in Exam 2 composed of problems E2P1 till E2P8 are shown in Figure 1. Even though two problems may have the exact concept count, their lines of code may differ based on the program plan for the solution. Some problems may have additional statements requiring using a different set of operators and print statements, thereby adding the number of lines of code without increasing the concept count. For example, this was the case in problem E2P4 compared to other problems with similar concept counts.

Appendix A shows the concept mapping of these problems, and Appendix B describes a partial list of the problem statements. The concept count of the Exam 2 problems is relatively high compared to the expected lines of code in the solutions. For example, writing a simple for-loop requires knowledge of arithmetic and Boolean operators, appropriate use of variables and their scope, and the syntactic elements used to compose expressions and statements. Additional concepts are used to write statements that form the loop's actions.

Exam 3 requires students to write modular code using methods. Students must comprehend the questions and translate the requirements into code by writing the correct return type, arguments, and statements in the method's body. Depending on the problem's requirements, a method's body may require the application of concepts such as arithmetic and Boolean operations, decision structures, or loops. Therefore, questions in Exam 3 also included the concepts that constituted the problems in exams 1 and 2. Appendix A reveals some of the concepts involved in Exam 1 and 2 that were repeated in Exam 3. Figure 1 shows that Exam 3 questions E3P1, E3P2, E3P3, and E3P4 have code lines with high concept counts. The question E3P5, which carried the most points and concept count, required students to write a menu-driven application that repeated several if-else statements to direct the program based on user choices during execution time. Repeating the if-else statements added code lines that used the same concepts, and therefore, the concept count did not increase as much as the code lines did. The table in Appendix B describes problem E3P5.

The Pearson correlation results indicated a significant positive relationship between the lines of code per exam problem and the number of concepts ($r = .592$, $p = .005$). Therefore, for this study, the number of correct lines of code written by students could be used to gauge their latent conceptual knowledge. It is important to ascertain this positive correlation if the correct number of lines of code is to be used to measure student performance in the exams.

Student Performance

Student submissions were scored based on the percentage of the expected lines of code that were correct for each question. Tables 4, 5, and 6 show the average values of correctly written lines of code (or statements) for each question from the three exams. The tables also show the average percentage score per problem and the number of concepts in each question. The Pearson correlation results indicated a significant positive relationship between the percentage of correct lines of code for each solution and the number of concepts used to solve the problem ($r = .67$, $p < .001$).

Problem	**Avg Score**	**# Concepts**	**Avg Correct lines of code**
E1P1	71.41%	9	0.714
E1P2	85.71 %	10	0.857
E1P3	52.38 %	9	0.524
E1P4	68.57 %	12	2.744
E1P5	96.03 %	11	5.76
E1P6	75.24 %	12	7.52
E1P7	58.57 %	10	9.376
E1P8	90.71 %	14	13.605
Avg values	**75.00%**	**10.875**	**5.1375**

Table 4 Exam 1 Results

Problem	**Avg Score**	**# Concepts**	**Avg Correct Lines of Code**
E2P1	82.86 %	11	2.487
E2P2	76.19 %	11	6.858
E2P3	73.33 %	12	2.199
E2P4	75.24 %	12	12.032
E2P5	77.14 %	12	0.771
E2P6	91.43 %	12	0.914
E2P7	86.67 %	12	3.468
E2P8	88.1 %	12	2.643
Avg Values	**79%**	**11.75**	**3.9215**

Table 5 Exam 2 Results

Problem	**Avg Score**	**# Concepts**	**Avg Correct Lines of Code**
E3P1	86.39 %	13	6.048
E3P2	73.47 %	15	3.675
E3P3	85.71 %	15	4.285
E3P4	81.63 %	15	8.976
E3P5	87.66 %	19	22.802
Avg values	**82%**	**15.4**	**9.1572**

Table 6 Exam 3 Results

Tables 4, 5 and 6 indicate that problems that required students to apply more concepts were the ones that students tended to score the most. In addition, the tables show that the average test score percentage increased after every exam. If the number of concepts used in a problem indicates its complexity, these results mean students are getting better at handling many concepts as they progress through the course.

Looking at the mapping of concepts in Appendix A, one can observe that Exam 1 uses many foundational concepts that are re-applied in all subsequent exams. Students also revisit many of these concepts in the assignment problems that follow Exam 1. Therefore, many code lines in the second and third exams repeat the concepts used in the first exam. Appendix A also shows that many of the same concept cluster together in various exam problems. For example, almost all programs use variables, data types, operators, and expressions. Knowledge of the correct syntax to construct statements is fundamental to all problems. Introducing newer concepts, such as decision structures and loops, helps to reinforce the use of foundational concepts covered earlier in the course, such as Boolean expressions and the use of different types of operators. The use of basic operators and inputs and outputs methods reoccur in almost all the programs that involve decision structures and loops. Therefore, repeated application of these concept clusters to meet the problem's sub-goals and create a more extensive program plan allows students to write correct code involving these concepts in subsequent exams.

A solid understanding of basic concepts and how they occur as a cluster to meet the program's goal and sub-goals allows students to incrementally integrate newer concepts successfully as they learn to write more extensive and complex programs.

5. DISCUSSION

It may appear concerning that students must grasp as many as 11 concepts by Exam 1, conducted during week 6 of the course. The Table in Appendices A and B shows that even writing a simple statement to solve a Boolean or Arithmetic operation requires knowledge of a cluster composed of many concepts. For example, nine concepts in the first three questions of Exam 1(E1P1, E1P2, and E1P3) are written using a single line of code that applies operator precedence rules. Even though there is no drastic increase in the number of concepts elsewhere in the course, students must learn to integrate newer concepts with what they already know to write programs as the course progresses.

Results from Tables 4, 5, and 6 indicate that students could write more correct lines of code as the semester progressed. Therefore, reapplying the same concept cluster many times throughout the semester and the familiarity gained would have led to mastery and better performance later in the course. However, it is to be noted that learning to solve different application problems happens not just by repeated exposure to the application of concepts but through the dynamic process of reconfiguring prior concepts to integrate a new concept required to solve a problem. Therefore, instruction could be designed to support acquiring new conceptual knowledge by reconfiguring and reapplying prior knowledge and skills and learning to apply a newer concept.

The progression of code writing exercise problems could play an essential role in helping students learn how to restructure their problem-solving schema and recombine prior concepts to solve new problems. For example, as evident from the assignment problem types in Appendix D, basic arithmetic operators, covered early in the semester, could be applied to develop various types of calculators. However, in the later module, problems that incorporate checkers into calculators will require students to incorporate Boolean operators into their pre-existing arithmetic operators and expressions schema. Problems that students solve later in the course require them further incorporate basic knowledge of arithmetic and Boolean operators in new ways to implement decision structures and loops. While there is considerable repetition of concept clusters throughout the course, there is also a need to restructure the previously learned cluster of concepts in new ways to solve different types of problems.

The problem-solving schema transferred through instructional code walkthrough helps students re-configure and re-apply concept clusters to solve problems. The instructional code walkthrough could help students identify the goals and sub-goals of the problem and then identify and configure a concept cluster to meet the sub-goals. For example, Appendix C shows the goals and sub-goal identification for a simple PIN identification problem, one of the assignment problems given to students. This solution pattern for the PIN identification problem could be modified to create applications that may validate user inputs or allow users to log in with a username and password. This problem and solution pattern could also be extended using loops to incorporate reattempts to check the user inputs.

Classic works in learning theory have argued that learners accumulate schema, or a problem-solving plan, rather than build their solution from scratch by applying all the elementary concepts (Rist, 1989; Clancy & Linn, 1999). Per this model, errors in applying a schema to solve a problem in an unfamiliar context or modifying the schema to fit the problem requirements may reveal flaws in the learner's understanding of the concepts used to compose the solution. The application of schema theory to computing pedagogy has taken a renewed interest, as indicated by the data-mining efforts to study common error patterns encountered by students during their learning process (Zehetmeir et al., 2016).

Repeated schema application or its modification to the problems' contexts allowed students to re-apply a cluster of concepts and assemble their solutions multiple times throughout the course. For example, based on the instructor's report, decision structure problems E1P5, E1P6, and E1P8 were analogous to problems in previously graded assignments. However, problems E1P1, E1P2, and E1P3 were single-line problems that did not directly resemble any assignment problems or were applied as part of a larger program plan. Even though students would have used smaller Boolean expressions to build decision structures, problems E1P1, E1P2, and E1P3 needed students to reason about the solution by considering every concept in the statement. Problem E1P7 was another problem that required a considerable modification of the assignment schema. Similar results were observed in Exam 2, where students scored the most if they could successfully identify similar problems from instructional code walkthroughs and assignments and transfer the schema to solve the exam problem.

Students scored the most in Exam 3 because the code inside the bodies of the methods repeated and reconfigured several code schemas previously covered in the assignments and exams. For example, the problems in the final exam required students to apply loop or decision structures in the body of the methods. Students could successfully write the body of the methods in Exam 3 if they learned how to integrate the method concepts with the problem-solving schemas used to solve loops or decision structure problems earlier in the course. A solid application of schemas within the body of a method allowed them to score partial points for a problem, even if they made mistakes directly related to the concept of a method, such as writing the method header or providing a correct return statement.

6. CONCLUSIONS

The result of this study indicates that programming is a cumulative skill and that as the semester progresses, students learn to write conceptually complex lines of code by accumulating and integrating many concepts into their solutions. The course starts with a high initial number of concepts and progresses with a relatively gradual increase of newer concepts that must be integrated with previously learned concepts. Integrating newer concepts to solve application problems also provides means to reconfigure code patterns and master the base concept clusters applied earlier in the course. Instructional code walks through, and practice assignments support the acquisition of programming skills by repeatedly integrating newer concepts into a cluster of concepts that appear in past assignments and exams.

This study confirms a positive correlation between the error-free lines of code produced for a solution and the number of concepts that students need to integrate to produce a solution. An explanation of why students can write correct lines of code despite increasing the conceptual complexity of the solutions is that they can learn the problem-solving schema and apply code patterns involving concept clusters. Students and instructors cope with an extensive concept count by clustering the concepts into code patterns corresponding to a problem schema. This study's finding has implications for designing instructional activities to help students recognize the instructor's problem-solving schema that deals with clusters of concepts that could be re-configured to meet a goal. Students may then remember each concept in isolation due to its meaningful association with other concepts in a solution's code pattern.

This study primarily focused on the ability of students to solve exam problems like those used during the instructional process. Future studies could investigate the complexity of exam problems by characterizing the concept clusters that appear in various problem-solving schemas. In addition, studies could be conducted to learn how students transfer problem-solving schema to unfamiliar problems. Finally, the difficulty of exam problems could be assessed based on not just the concept count but also conditioned on prior exposure to similar problems.

7. REFERENCES

Bennedsen, J., & Caspersen, M. E. (2019). Failure rates in introductory programming. *ACM Inroads*, *10*(2), 30–36. https://doi.org/10.1145/3324888.

Berssanette, J. H., & de Francisco, A. C. (2022). Cognitive Load Theory in the Context of Teaching and Learning Computer Programming: A Systematic Literature Review. *IEEE Transactions on Education*, *65*(3), 440–449. https://doi.org/10.1109/te.2021.3127215.

Braarud, P. I. (2001). Subjective Task Complexity and Subjective Workload: Criterion Validity for Complex Team Tasks. *International Journal of Cognitive Ergonomics*, *5*(3), 261–273. https://doi.org/10.1207/s15327566ijce0503_7.

Cherenkova, Y., Zingaro, D., & Petersen, A. (2014). Identifying challenging CS1 concepts in a large problem dataset. *Technical Symposium on Computer Science Education*. https://doi.org/10.1145/2538862.2538966.

Clancy, M. J., & Linn, M. C. (1999). Patterns and pedagogy. *The Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education*. https://doi.org/10.1145/299649.299673.

DeFranco, J. F., & Voas, J. (2022). Revisiting Software Metrology. *Computer*, *55*(6), 12–14. https://doi.org/10.1109/mc.2022.3146648.

Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. *Psychological Review*, *102*(2), 211–245. https://doi.org/10.1037/0033-295x.102.2.211.

Ford, M., & Venema, S. (2010). Assessing the Success of an Introductory Programming Course. *Journal of Information Technology Education: Research*, *9*, 133–145. https://doi.org/10.28945/1182.

Goldman, K., Gross, P., Heeren, C., Herman, G. L., Kaczmarczyk, L., Loui, M. C., & Zilles, C. (2010). Setting the Scope of Concept Inventories for Introductory Computing Subjects. *ACM Transactions on Computing Education*, *10*(2), 1–29. https://doi.org/10.1145/1789934.1789935.

Harland, J., D'Souza, D., & Hamilton, M. (2013). A Comparative Analysis of Results on Programming Exams. *Proceedings of the Fifteenth Australasian Computing Education Conference (ACE2013).*

Kramer, M., Barkmin, M., Tobinski, D., & Brinda, T. (2017). Understanding the Differences Between Novice and Expert Programmers in Memorizing Source Code. *IFIP Advances in Information and Communication Technology*, 630–639. https://doi.org/10.1007/978-3-319-74310-3_63.

Meyer, J.H.F. & Land, R. Threshold concepts and troublesome knowledge (2005): Epistemological considerations and a conceptual framework for teaching and learning. *High Educ 49, 373–388*.

Medeiros, R. P., Ramalho, G. L., & Falcao, T. P. (2019). A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. *IEEE Transactions on Education*, *62*(2), 77–90. https://doi.org/10.1109/te.2018.2864133.

Muller, O., Ginat, D., & Haberman, B. (2007). Pattern-oriented instruction and its influence on problem decomposition and solution construction. *Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*. https://doi.org/10.1145/1268784.1268830.

Petersen, A., Craig, M., & Zingaro, D. (2011). Reviewing CS1 exam question content. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*. https://doi.org/10.1145/1953163.1953340.

Rist, R. S. (1989). Schema creation in programming. *Cognitive Science*, *13*(3), 389–414. https://doi.org/10.1016/0364-0213(89)90018-9.

Sanders, K., & McCartney, R. (2016). Threshold concepts in computing. *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*. https://doi.org/10.1145/2999541.2999546.

Schulte, C., & Bennedsen, J. (2006). What do teachers teach in introductory programming? *Proceedings of the Second International Workshop on Computing Education Research*. https://doi.org/10.1145/1151588.1151593.

Sheard, J., Simon, Carbone, A., Chinn, D., Laakso, M. J., Clear, T., de Raadt, M., D'Souza, D., Harland, J., Lister, R., Philpott, A., & Warburton, G. (2011). Exploring programming assessment instruments. *Proceedings of the Seventh International Workshop on Computing Education Research*. https://doi.org/10.1145/2016911.2016920.

Sheard, J., Simon, Carbone, A., Chinn, D., Clear, T., Corney, M., D'Souza, D., Fenwick, J., Harland, J., Laakso, M-J., & Teague, D. (2013). How difficult are exams? A framework for assessing the complexity of introductory programming exams. In A. Carbone, & J. Whalley (Eds.), *Proceedings of the Fifteenth Australasian Computing Education Conference (ACE 2013): Adelaide, Australia, 29 January - 1 February 2013* (Vol. 136, pp. 145 - 154). (Conferences in Research and Practice in Information Technology (CRPIT); Vol. 136). Australian Computer Society Inc. http://crpit.com/confpapers/CRPITV136Sheard.pdf

Simon, D'Souza, D., Sheard, J., Harland, J., Carbone, A., & Laakso, M. J. (2012). Can computing academics assess the difficulty of programming examination questions? *Proceedings of the 12th Koli Calling International Conference on Computing Education Research - Koli Calling '12*. https://doi.org/10.1145/2401796.2401822.

Tew, A. E., & Guzdial, M. (2010). Developing a validated assessment of fundamental CS1 concepts. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. https://doi.org/10.1145/1734263.1734297.

Watson, C., & Li, F. W. (2014). Failure rates in introductory programming revisited. *Proceedings of the 2014 Conference on Innovation &Amp; Technology in Computer Science Education - ITiCSE '14*. https://doi.org/10.1145/2591708.2591749.

Zur, E., & Vilner, T. (2014). Assessing the assessment-Insights into CS1 exams. *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. https://doi.org/10.1109/fie.2014.7044330.

Appendix A
Exam problem Concepts

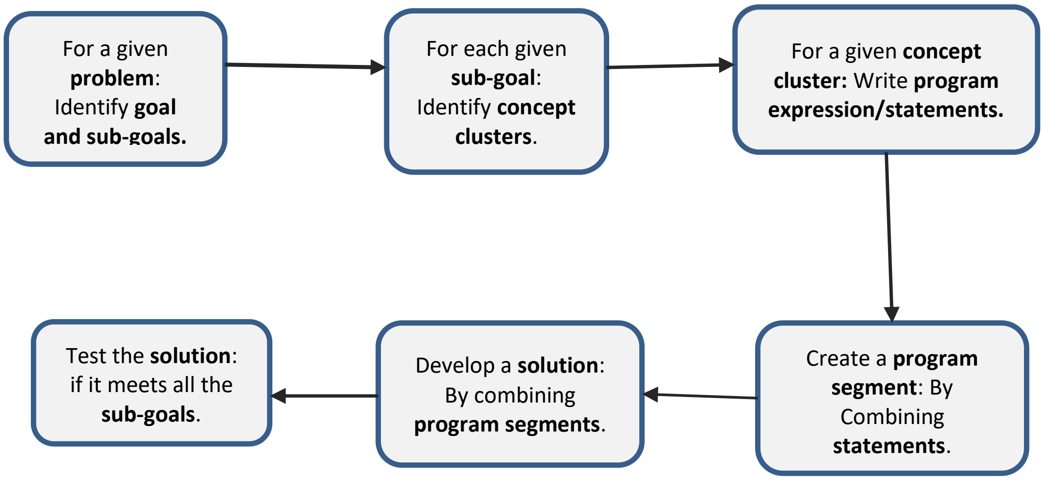
	Exam 1								Exam 2								Exam 3				
Concepts	**P 1**	**P 2**	**P 3**	**P 4**	**P 5**	**P 6**	**P 7**	**P 8**	**P 1**	**P 2**	**P 3**	**P 4**	**P 5**	**P 6**	**P 7**	**P 8**	**P 1**	**P 2**	**P 3**	**P 4**	**P 5**
Method Structure																	x	x	x	x	x
Method Parameter																	x	x	x	x	x
Method Return																	x	x	x	x	x
Method Call																					x
Syntax	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Data Types	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Variables	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Literals	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Boolean Operators	x	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x
Arithmetic Operators		x			x	x		x	x	x	x		x	x	x	x	x				x
Assignment Operator	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Operator Precedence	x	x	x	x		x		x													x
Expression	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Statements	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Conditionals				x	x	x	x	x			x			x				x	x	x	x
Decision Structures					x	x	x	x											x	x	x
Loops									x	x	x	x	x	x	x	x					
Nested Loops																x	X	x			
String methods				x								x						x			
Variable Scope													x	x	x			x	x	x	x
Print Methods							x	x	x		x	x	x		x	x			x		x
Input(Scanner) Methods				x				x		x		x						x		x	x

Concepts used in each exam problem for the three exams. The grey cells indicate the main course concept/topic that is evaluated in the problems. Cell that are marked with an 'x' indicate the concept that is used in the lines of code of a correct solution.

Appendix B
Some of the Exam Problems from Exam 1, 2 and 3

<table>
<tr><th>Exam Problem</th><th>Problem/ Question</th></tr>
<tr><td>E1P1</td><td>What will be the value of boolean var1, which is given as : var1 = (((a*b)<=5)&&(b<1)); if you substitute a = 2 and b = 1 ?</td></tr>
<tr><td>E1P3</td><td>What will be the value of boolean var1, which is given as : var1 = (a.equals("Apple")); if you substitute a = "apple"</td></tr>
<tr><td>E1P5</td><td>Given two input variables : double length and double width. Given one output variable : double perimeter.
Assume that values of length and width are already obtained. Write if statement (just the if statement(s) with the action performed , not the entire program) for the following conditions:
Check if the dimensions are big as follows: If length is greater than 20.0 or width is greater than 20.0, give an output to tell the user that the dimensions are too big.
Check if the dimensions are small as follows: if the length is less than 5.0 or width is less than 5.0, give an output to tell the user that the dimensions are too small.
If the dimensions are neither too big or too small, based on the above two checks - tell the user that the dimensions are within the proper range of values . Then, calculate the area = length * width and print the value of that perimeter.</td></tr>
<tr><td>E1P8</td><td>Write a program that obtains from the user the age of a child in months. The program will determine the required next vaccinations based on the given age. Write a complete program that will look at the given age in months and determine the next vaccination required. You may skip commenting your code for this exam to save time. You program should compile and be logically and syntactically correct.
<table>
<tr><th>Baby's Age in months</th><th>Next Vaccination</th></tr>
<tr><td>0</td><td>HepatitisB</td></tr>
<tr><td>Greater than 0, less than 6 (inclusive)</td><td>DaTP</td></tr>
<tr><td>Greater than 6, less than 12 (inclusive)</td><td>MMR</td></tr>
<tr><td>Greater 12</td><td>"Will complete the program later on"</td></tr>
</table></td></tr>
<tr><td>E2P2</td><td>In this problem you will write a loop in which you ask users to enter the price of an item and add that price to the total price. In your loop, you will ask the user to enter a 1 to “Scan” and a 2 to “quit”. Loop until the user types 2 - to quit.
For as many times, as the user enters a 1- to "Scan an item' : ask the user for the price of the items , and add this price to a variable called totalPrice.
The variables price and totalPrice are both doubles .Assume that the Scanner object is already declared and named as input.
Declare any extra variables that you have used in your loop - other than price, totalPrice or input.</td></tr>
<tr><td>E2P6</td><td>The for loop shown below loops several times and produces a final value of i that is used to calculate the value of j. However, there is an error : int j = i % 5 ; //This statement shows an error : "cannot find symbol i"
Errored code:
for(int i = 1; i<100; i=i*5){
System.out.println (i);
}
Rewrite the code above so that it fixes the error given in the error statement</td></tr>
<tr><td>E3P3</td><td>Define/Write a method called codeThePlayer that takes two argument – an integer called playerID and a String called playerName . This method has a void return. If the playerID value is equal to 100, the method prints out the following : "Admin ID ". Else, if the playerID is not a 1, the method prints out the playerName, followed by the statement: "Not Admin".</td></tr>
<tr><td>E3P5</td><td>Menu driven program
Write a program called pointsCalculator that calculates the total points earned by using a credit card for travel and hotel stays.
The program provides the user with the following menu :
“ Enter 1 to select mileage points”
“Enter 2 to select a hotel points”
“Enter 99 to quit”
If the user enters a 1 , ask the user to enter the mileage (which will be a double type). Scan the mileage and call a method called calculateMileage that takes in as mileage as a parameter. This method returns a double value to be stored in a variable called mileagePoints. Print out the value of mileagePoints.
If the user enters a 2, ask the user to enter the number of hotel stays(which will be an integer type). Scan the hotel stays and pass this variable as a parameter to the method calculateHotelPrice . This method returns a double variable called hotelPoints. Print out the values of hotelPoints.
Assume the methods are already defined - you just need to call them in the code. You also don't need to implement a while loop.</td></tr>
</table>

Appendix C
Problem-Solving Steps/Schema Breakdown Used for Instructional Code Walk Through



Problem	Checker - Simple PIN validation
Statement	Write a program that validates a user based on a 4-digit PIN value. The program allows the user to input their 4-digit PIN and compares this value with the value on file. Assume that the PIN value stored in the file has been obtained and stored in variable in your program.
Goal	Validate the user's PIN value with the PIN value on file and take appropriate actions for pasisng and failing validation.
Sub-goal 1	Input: Program should obtain the PIN value from the user.
Sub-goal 2	Compare for equality : Program should compare the PIN value with the PIN value on file.
Sub-goal 3	Action 1: Output an validation message if the PIN values matches / validation passes.
Sub-goal 4	Action 2: Output an error message if the PIN values do not match/ validation fails.
Sub-goal 1 program segment	a. Declare a variable to store the user PIN - [Data type,variable,assignment operator,initial value/int literal,expression,statement] : int userPIN = 0; b. Prompt the user for user PIN value - [Data type,String literal] : "Enter the 4-digit user PIN " c. Write a print statement- [String literal,method,statement]: System.out.println("Enter the 4-digit user PIN "); d. Scan and save the user input from the console - [Scanner method,data type,assignment operator,variable,datatype, expression,statement]: userPIN =
Sub-goal 2 program segment	a. Declare file PIN variable with a value of 6788 and save the value in that variable - [Data type,variable,assignment operator,expression,statement]: int filePIN = 6788; b. Boolean Expression for comparison -[Comparison operator, data types,variables, expression] : userPIN == filePIN c. If statement using comparison operator: [Decision structure, comparison operator, data types, variables] : if(userPIN == filePIN){..} else {...}
Sub-goal 3 Program segment	a. Create a validation message- [Data type/String literal] : "You are validated." b.For the if condition print the validation message- [String literal, method, statement] : if(userPIN == filePIN) {System.out.println("You are validated.");}
Sub-goal 4 program segment	a. Create an error message- [Data type/String literal] : "PIN Error, PIN not valid." b. For the else condition print the validation message- [String literal, method, statement] : else { {System.out.println("PIN Error, PIN not valid.");}

APPENDIX D
Assignment Problems Categorized by Problem Types and Concept Clusters

List of Application Problem Patterns in Assignments	
Module 0- Variable, Data types, Scanner methods	**Calculators - Data types, Arithmetic Expressions, Input/Output, Strings**
	Shipping Cost
	Taco Price
	TypeCasting Inputs
	Flooring Cost
	HealthData
	Make Change
	Checkers: Data types, Boolean expressions, Input/Output, Strings, IF/ELSE
Module 1 - Decision Structures	Find Special Values : filtering out special values from a series of input data
	Range Checker - identifying the range of a given input value
	Age Checker - identifying the age group that a person falls under
	PIN validation - validating user PIN value (without retries)
	Checkers +Calculators: Data types, Boolean/Arithmetic Expressions, Input/Output, IF/ELSE (mutliple ifs, else if)
	Score Difference - figuring out game winners based on score differences
	Age Checker - Binning for creating Histograms
	Age & ZipCode checker - Demographic categorization problem
	Year To Century Converter
	Ticket Price Based on age /product type / discount code
	Electrice Power Consumption Calculator for multiple home appliance types.
	Decisions&Policies: Boolean/ Arithmetic Expressions, Input/Output, nested IF/ELSE, Strings
	Rock Paper Scissor Game
	Labor Charge Calulator for Lawn Service
Module 2 - Loops	**Counters: Data types Boolean/Arithmetic Expression, Input/Outputs, Strings, Loops (while and for)**
	Shopping Data Input till user wants to quit using sentinel value
	PIN Validation with retries
	Interest Calculator
	DivideByTwo series generator
	ABCounter ForLoop Implementation
	ABCounter WhileLoop Implementation
	FutureTuition Calculator with inflation rates
	InsectGrowth - series generator with varying parameters
	Loops with Decisions - Data types,Boolean and Arithmetic expression, Input/Output, Strings, IF/ELSE , Loops
	Validating Inputs - with infinite retries.
	Menu Driven Application with multiple rounds of entry
	Password, Username validation with re-tries and lockout
Module 3- Methods	**Methods - Return types, parameters, Data types,Boolean, Arithmetic expressions, Input/Output, Strings, IF/ELSE, Loops,**
	Print Shapes : Methods to print different shapes without parameters
	Customaisable Face Printer with parameters - Methods to print different types of faces, with parameters
	Dinner Price Calculator1 - Methods for user input, entrée price, discount calculator
	eBayFee: Methods for identifying user types, fee calculation, output
	Length Convertors : Methods for multiple types of conversions
	Ticketing Application 1: Methods for user input, calculation
	Ticketing Application 2 - Methods for user input, decision making, calculation
	Dinner Price Calculator2 - Menu driven app, methods for user input, entrée price , drink price, receipt
	TaxApplication - Methods for: user inputs, output display, calculator and category checkers

Implementing a First-Year Experience Course for IT Majors

David M. Woods
woodsdm2@miamioh.edu
Computer & Information Technology Department
Miami University Regionals
Hamilton, OH 45011, USA

Abstract

Schools have implemented the high-impact practice of first-year experience (FYE) courses to assist students with the transition to college. These courses help connect new students to the school with the goal of improving retention. While students in computing majors face the same challenges as other new college students, they also face some challenges specific to computing fields. Additionally, computing fields face ongoing concerns about enrollment and diversity. This work discusses the implementation of an information technology (IT) specific FYE course combining content from a university-wide FYE course with content focused on helping students gain a better understanding of the IT curriculum and IT career paths.

Keywords: First-year experience, student success, pedagogy.

1. INTRODUCTION

The transition to college, whether it be from high school, military service, or work, is a challenging time for students. Successful transition into college requires students to build connections to the college, their chosen major, and their peers. Students also need to develop a clear goal for why they are attending college to motivate the effort required to succeed in college. The first-year experience (FYE) course has been identified as a high-impact learning practice that can support the transition to college (Kuh, 2008).

While a successful transition to college is essential for all students, several factors can make this more challenging for computing students. High school students will have taken several years of science, math, social studies, and English but have had fewer or possibly no opportunities to take computing courses. This may make students hesitate to choose a computing major or give them a limited view of what computing involves. This limited exposure to computing and the rapidly changing nature of technology can make it hard for students to develop a clear career goal.

The fortunate coincidence of the introduction of a new university-wide first-year experience (FYE) course at the university at the same time as a redesign of the introductory Information Technology (IT) course sequence in the Computer and Information Technology (CIT) department provided an opportunity to consider how to combine these efforts into a focused approach to helping students successfully transition into college and the CIT program.

2. BACKGROUND

There is extensive research on first-year experience courses in general and efforts to improve the enrollment and retention of computing students through focused efforts in first-year courses.

First-Year Experience Courses

Courses designed to help address the challenges new students face when entering college have a long history (Ryan & Glenn, 2004) and many names. These courses are part of a larger effort designed to transition students to a successful college career. Other parts of this effort include new student orientation programs, academic advising, and all of a university's student engagement and support services.

There are three main formats used for FYE courses (Ryan & Glenn, 2004). The extended-orientation model (also referred to as academic socialization) focuses on integrating the students into the university community. This format focuses on the history of the college, available services, and engaging with other students. This model is traced to an effort introduced at the University of South Carolina in 1972 as part of a response to campus riots in 1970 (History, n.d.).

A variation on the socialization approach is a course organized around an academic theme. This format was supported by the Boyer Report (Kenny & Boyer, 1998), which suggested that the "first year of a university experience needs to provide new stimulation for intellectual growth" (p. 19).

A final FYE course format focuses on preparing students for success by teaching study skills and learning strategies. In addition to the value of these skills, teaching these skills also offers an opportunity to promote metacognition (Hattie, Biggs, & Purdie, 1996).

In reality, the FYE program at any school will be a unique mix of the different approaches. FYE programs at different schools will have different goals based on the demographics and needs of the school's student population, the credit hours allocated to the course, and many other factors. For example, open enrollment institutions and institutions with a larger population of students returning to education may focus more on study skills and learning strategies (Mayo, 2013).

Several authors have looked at the impact of FYE programs, including the impact on retention, academic performance, and graduation rates. The impact on retention is particularly relevant since 30% of dropouts occur in the first year (Hanson, 2021). Research finds positive impacts on retention, student performance, and other factors (Lang, 2007; Erickson & Stone, 2012; Bers & Younger, 2014; Howard & Jones, 2000). Ryan and Glenn (2004) report on a comparison of the academic socialization and learning strategy formats and find that the learning strategy format had a positive impact on retention while the academic socialization format had no impact on retention.

Discipline-Specific FYE Courses
In addition to the extensive discussion of university-level FYE programs, there is also research on discipline-specific FYE efforts. Computing disciplines face specific challenges in retaining students and promoting diversity that could be addressed in an FYE course.

While recent enrollment in computing fields shows continued growth, future trends are unclear (National Academies, 2018). In addition, while computing enrollments have grown, the field remains "one of the least diverse disciplines in terms of the representation of women and underrepresented minorities, both in higher education and in the workforce." (National Academies, 2018, p 4). A recent ACM work on retention reinforces this with the observation that "the demographics of the students involved in computing remain stubbornly consistent" (Stephenson et al., 2018, p i). This work provides a lengthy set of recommendations, including overcoming misconceptions about computing fields and researching new interventions.

Several studies of computing-specific FYE efforts have found positive impacts on retention (Ott, 2014; Mathis, 2008; Reese et al., 2013; Albarakati, DiPippo, & Fay-Wolfe, 2021) through a variety of approaches. Many of these efforts combined computing content with practices suggested for general FYE courses. For example, Albarakati et al. (2021) added course activities designed to help students develop a sense of belonging and promote social engagement. DeClue et al. (2011) report on similar activities, including service projects to promote a sense of community.

To address diversity concerns, it is helpful to look at why students choose computing majors in the first place. Compared to other majors, computing fields face many challenges in attracting students. While most high school students will take multiple courses in math, science, history, and other long-established fields, the percentage of high school students taking computing courses is in the single digits in many states (Guzdial, 2019).

Another issue is limited awareness of what computing professionals actually do. When Carter (2006) surveyed high school students, the top reason for not choosing a computing major was not wanting to sit in front of a computer all day. The top reason for selecting a computer major was also illustrative. For men, the top reason was an interest in computer games, while for women, it was "their desire to use it in another field" (Carter, 2006, p. 29). Related to this is the issue of understanding the different computing fields.

Connolly et al. (2016) examine this topic and provide suggestions about advising students in selecting the computing field that best matches their interests.

There have been several efforts to help first-year computing students better understand the wide range of computing careers and improve retention. In one approach, an “Explorations in Computing” course was revised to combine faculty and industry presentations on different facets of computing, one-on-one student-alumni interaction, and the development of professional skills such as resume writing (Ott, 2014). After the revision, the authors found that students were more engaged in the class, with more students completing assignments, higher grades, and fewer students dropping the course. They also found more engagement with a mid-semester career fair.

In another effort, a computing fundamentals course was revised to address the characteristics of current Gen Z students (Robertson, 2011). The breadth of course topics was reduced to allow time to provide more context to connect the remaining topics to real-world examples, and a lecture-based class format was replaced with a discussion-based format. In addition, career discussions were added, and a discussion of related careers was part of each topic module. These changes resulted in more engaged students, improved attendance, better grades, and increased retention.

Another interesting approach combines the FYE approach based on study skills and learning strategies with technical content as part of a plan to promote the development of non-technical skills to support the achievement of technical learning outcomes (Humpherys, Babb, & Abdullat, 2015). Initial efforts in a first-year programming course have led to a 10 % increase in student persistence.

3. COURSE DESIGN

Design Considerations

Prior to the redesign discussed in this work, the introductory IT course sequence consisted of two three-credit courses designed to be taken during a student’s first year in the program. Both courses covered various technical topics designed to provide a general introduction to IT and foundational material for later courses in the curriculum. Over time, the two courses had evolved as individual instructors added or changed content, leading to overlaps and gaps due to the failure to coordinate the changes to the two courses. In addition to supporting associate’s and bachelor’s degrees in IT and a bachelor’s degree in Health Information Technology (HIT), these courses were also part of an IT minor and an IT concentration that students from other majors could complete to satisfy part of the university’s general education requirement.

Discussions about redesigning the courses had started before the university-wide FYE course requirement was adopted. The primary considerations for redesigning the courses were to update the content to remove older technologies and add newer technologies such as cloud computing, while improving the alignment with later courses in the curriculum. There was also interest in adding content to help students learn more about IT majors and careers early in their academic careers to allow students to consider a change of major before investing a lot of time and effort in a major that doesn’t match their interests. A final motivation was to work on developing non-technical skills to address the concerns identified by Carter (2006) and Humpherys (2015) early in the curriculum. Additional considerations included reducing the two courses into a single course to accommodate a state-mandated reduction in the total credits for a bachelor’s degree and simplifying the processing of credits for transfer students entering the program.

When the university announced that all incoming students would be required to take a 1-credit UNV 101 course, the CIT department saw an opportunity to expand this to include computing-specific content, including exploration of computing careers and academic advising specific to the CIT program requirements. In addition, the department felt that a computing-specific FYE course would allow the introduction of IT-specific content for the study skills content. For example, introducing Agile practices and tools for time management. Also, the one-credit UNV 101 course plan included having faculty from all departments teach sections of the course. Since most courses in the CIT department are three-credit courses, this would have complicated assembling a full 12-credit teaching load for faculty.

The solution developed included a two-credit computing-specific version of the UNV 101 course and a four-credit course that would introduce students to various technical topics. Ideally, the same faculty member would teach both courses to simplify faculty scheduling and provide students with a connection between the two courses. The two courses also form the basis for a campus-level "course-in-common" effort based on research suggesting that learning communities can increase student motivation and retention (Cabo & Satyanarayana, 2018; Saulnier, Brooks, Ceccucci, & White, 2007).

An additional goal of the introductory course sequence was to ensure that students were in the correct major. The university asks incoming students to declare a major during new-student orientation, and experience has shown that these are not well-informed decisions in some cases. As discussed earlier, students may not have a good understanding of computing careers (Carter, 2006), or the choice of major may be made by a parent. The department felt that advising students to take the two courses simultaneously would allow them to gain experience with their major early in their college experience. Academic advising content in the FYE course could help students with concerns explore other majors. While this might lead to students leaving the IT major, it was expected that this early focus on ensuring students were in the right major would help retain the student at the university and limit potential delays in their path to graduation.

All students working towards an associate's or bachelor's degree in IT are required to take both courses. Given the content of the computing-specific FYE course, the department initially chose to avoid offering the course in an online format. The CIT department's Health Information Technology (HIT) program is only offered in a fully online format, so these students are not required to take the IT-specific FYE course. However, HIT students are made aware of the FYE course, and HIT students from the local area often enroll in the course.

Course Details
The IT-specific FYE course is titled "IT People and Practices." This course builds on the university-wide UNV 101 FYE course's goal of connecting new students to the university with the additional goals of connecting students to their chosen major and connecting students to their future careers.

The learning outcomes for the IT-specific FYE course can be found in Appendix A. These build on the university-level course by adding outcomes designed to help students learn about IT careers and the knowledge and skills needed to be a successful IT professional.

The design of the IT-specific FYE course is based on the design of the university-wide UNV 101 with two main changes, the addition of IT-specific content and modifications to reflect a different student population. The CIT department is located on the university's regional campuses, not the main campus. The main campus is a selective admission, residential campus, while the regional campuses are open-enrollment, commuter campuses. These differences mean that regional campus students are more likely to be first-generation college students, include more non-traditional age students, are commuter students, and often mix work and school.

The four-credit technical introduction course titled "IT Tools and Techniques" is broken into four three-week-long content modules and a short introductory module. The four major topic areas covered include:

- Hardware infrastructure, operating systems, and cloud computing.
- Web programming – HTML, CSS, and JavaScript.
- Networking
- Data and Databases

These four topics were selected to provide a broad overview of fundamental IT topics and for their support for later courses in the IT curriculum. The introductory module provided an introduction to the course. The introductory module also covered foundational topics used in the other modules, including binary, data representations, and digital logic.

Following suggestions from the literature (Roberson, 2011), each of the four major topic areas included context to help students understand how the topic was present in their daily, technology-enabled life and concluded with a discussion of IT careers that would use the skills students learned in the module.

4. COURSE IMPLEMENTATION

Course Content
The starting point for content in the IT-specific FYE course is the content from the university-wide UNV 101 course. The UNV 101 course uses a mix of the extended orientation and study skills approaches discussed previously. The content contains material to help students transition from high school to college, along with content to build skills for success at the college level.

The content to support the transition from high school to college includes a discussion of the differences and an activity to develop a personal vision for what they hope to achieve by attending college. This content contains introductions to campus support services, including the tutoring center, library services, and student organizations. Given the student population of the regional campuses, this content is expanded to consider students who are not coming directly from high school.

The content to build college success skills covers many areas, including time management, developing a growth mindset, critical thinking, financial literacy, and personal care – both physical and mental. For the regional campus student population, suggestions from student affairs staff have led to expanded content on financial literacy and personal care. In the IT-specific FYE course, this area provides an opportunity to introduce skills seen in the IT environment. For example, the discussion of time management offers the opportunity to introduce several Agile practices.

The IT-specific FYE course extends the UNV 101 course into a two-credit course that is typically offered in a hybrid format with one fifty (50) minute class meeting per week, along with additional online reading and video content. During the COVID pandemic, the course was taught in an online synchronous format, and currently, the course is offered in both formats. Typically, there is demand for two or three sections of the course, with fifteen (15) to twenty-five (25) students in a section.

The IT-specific FYE course adds content in two additional and related areas. Appendix B contains a table showing the content and assignments from a recent section of the course. Content on computing professions helps students develop initial career plans, and content on the CIT curriculum helps students match their career plans to select the appropriate path through the department's curriculum. This content is introduced on the first day of class with a discussion of "What is IT?" and "What do IT professionals do?" Throughout the semester, several recent graduates return to speak on their current jobs, tips for success in college and the IT program, how they found their first IT job, and other topics. Students also research current Bureau of Labor Statistics (BLS) data (Bureau of Labor Statistics, n.d.) and explore local job postings to learn more about the wide range of career opportunities in IT. The concept of professional networking is also introduced, and students set up a LinkedIn profile and make connections with fellow students, department faculty, and guest speakers in the course.

To help students connect with the CIT program curriculum, an entire class period is spent reviewing the available CIT degrees and discussing the detailed requirements for each degree. University-level graduation requirements are also discussed. Students use a degree-specific template to start developing a graduation plan during this class. The department's chief advisor joins this class session to help students understand the degree requirements and work with students who have existing credits from previous college courses or military credits to ensure that these credits are correctly applied. This class session is timed to coincide with the beginning of the registration cycle for the following semester. Students build on this session by completing an assignment where they meet with their assigned departmental advisor to plan their courses for the next semester.

Course Pedagogy

In the IT-specific FYE course, course content is organized into weekly modules. Each week, students were provided an online agenda that outlined the topics for the upcoming week, any activities students needed to complete before class, resources related to the week's content, and all assignments that students needed to complete before the next class. To promote student reflection and develop time management skills, students must complete a weekly reflection assignment. Each week students write at least two paragraphs. First, students discuss the current week – what they learned in class, what went well for them, what didn't go as well, etc. In the second paragraph, students discuss what they hoped to learn in the following week and what they needed to be working on. The weekly agenda page was designed to be an easy source of information to support both aspects of the weekly reflection assignment.

Promoting exploration and reflection are key pedagogical practices throughout the course. Most course assignments ask students to explore a new concept that has been introduced in class and then reflect on how they can make use of the concept in the future. For example, after a class discussion about time management where students engage in the Agile-based personal Kanban process and discuss the Eisenhower decision matrix, they complete an assignment where the students identify ten tasks they need to complete in the next week and use a time management practice to organize the tasks. At the end of the week, students submit a document listing the ten tasks, a discussion of the tool they used, a reflection on how well the tool worked, and a discussion of their plans for improving their time management practices.

Another pedagogical practice used in the class was offering students choices to make the course more learner-centered (Becker 2006). This was done by setting up three "you choose" groups of assignments on student success, career planning, and information literacy. The assignments offered ways for students to learn about different campus services. For example, setting up a meeting with a peer tutor, having their resume reviewed by the career services office, or learning about library resources through a virtual escape room activity. Each group contained six or seven assignments with different values depending on the expected difficulty. Students could choose which assignments they completed, with a maximum point value that could be earned in each group. Depending on which assignments the student chose, a student needed to complete three to five of the assignments to earn the maximum possible points.

5. DISCUSSION

The IT-specific FYE course has now been taught every semester for five years. Most students in the fall semester offering are newly enrolled students. The spring semester has a mix of newly enrolled students, transfer students, and students who did not follow the advice to take the course in the fall semester. Discussions with upper-level and graduating students show that they saw a clear value from this course and also provided ideas for improving the course.

Efforts are underway to explore how the IT-specific FYE course has affected retention. The ongoing COVID-19 pandemic is affecting retention, so it is a challenge to measure the impact of the IT-specific FYE course. Additionally, since one goal of the course is to ensure that students are in the correct major, rather than retaining them in a CIT major, this will require access to a broader set of data. While analysis of this data is ongoing, it is possible to consider how students have done in the course and evaluate the course content.

Students

Looking at student engagement and success in the IT-specific FYE course, students fall into several groups. The largest group contains students who fully engage in the content and consider how to use what they learn to plan their future studies and career. Some students in this group were on this path at the start of the class due to good preparation in high school or knowledge shared by family members. Others in this group entered the class unsure of whether they belonged in college or were concerned about whether they could succeed in college. It is rewarding to see these students engage with the FYE class and build their confidence in their ability to succeed in college and their future IT career.

Another group of successful students is those who determine that an IT major is not for them. These students are successful with the non-IT content in the course but can struggle with the discussions about IT careers and the IT curriculum. Efforts are made to connect these students with campus resources that can help them explore other majors. Where possible, they are allowed to modify IT-specific assignments to fit their new major and career interests.

The less successful students fall into one of two groups. The first contains students who are partially or fully engaged in the class meetings but struggle to complete course assignments. These are students who are struggling to adapt to college. These students may struggle with the lack of structure in college classes compared to high school classes, poor time management, lack of study skills, lack of motivation, or issues including financial strain, mental health, and family issues. As soon as instructors notice students are struggling, the university's Academic Early Alert system is used to engage student support services to work with the students. In some cases, the additional support allows a student to complete the course successfully. Unfortunately, other students are less successful in dealing with their challenges during the short duration of the course. While students in this group may not successfully complete the course, they learn about the personal challenges they face and make valuable connections to campus support services.

The other group of less successful students contains students who choose not to engage in the class. Generally, these students don't see the value of the course. They think they already know how to be successful in class, dismiss the content since it is not what they consider a "real IT" class, or feel that courses not directly connected to their (often vague or impractical) career plans are a waste of time. Instructors make an effort to find a way to engage these students, and some choose to engage, but some continue in the class with limited engagement.

A final group of students should be mentioned. These are IT majors in their second and later years. When students cannot take the class in their first semester, advisors will find an alternative, but some students self-advise and take the course since it is listed as a requirement in the university bulletin. Some of these students are fully engaged and are great additions to the class, while others are minimally involved.

Content
Student feedback from an end-of-semester reflection assignment and discussions with individual students shows that students are engaged with all of the course content, but some content areas clearly work better than others.

The guest speakers are always the most popular aspect of the course. Instructors seek to schedule a diverse set of speakers, ideally allowing every student to find a speaker they relate to. Students find encouragement in seeing that the guest speakers also had concerns about succeeding in college and are receptive to the speakers' suggestions about study skills, the value of group work, how to find an internship, and many other topics.

Students also find value in the discussions of time management and personal wellness, and the students' weekly reflections regularly mention how they are using content from these areas. The content on financial literacy also gets a lot of positive comments. Students complete a basic budgeting activity and regularly mention being surprised at how much money they spend and set goals to increase their savings or reduce spending in specific areas.

Two areas of the course have mixed results. Students are very engaged in the content, but the submitted work needs improvement. One area is the discussion of jobs. In searching for job postings, they have trouble identifying jobs that match their interests or are entry-level positions. Part of the challenge here is that many job postings are hard to decipher. The other area with mixed results is an end-of-semester assignment where students record a 30-second elevator pitch. Students are enthusiastic about the activity, but their videos suffer from a lack of professionalism.

A couple of aspects of the class need improvement. One is the discussion about the CIT curriculum and the assignment to develop an initial graduation plan. This topic has several external problems, including a complex process and poor tools to support planning. It is also a challenge to get students to take ownership of this process since many high school students have limited choices for courses and may be used to others making decisions for them. Improving this aspect of the course is an ongoing effort.

Another challenge is some of the content from the university-level UNV 101 course. Staffing changes have led to gaps in communication and loss of knowledge about external tools used to support topics like diversity, equity, and inclusion that are not discovered until students report problems.

6. CONCLUSIONS

The IT-specific FYE course has been a positive addition to the CIT curriculum. Helping students identify their career goals and understand the department curriculum is an ongoing effort, but having a dedicated effort to address these during a student's first semester in college builds a good foundation that later courses in the curriculum can build on.

Additionally, identifying students who are in the wrong major allows these students to find the right major and reduces the number of students who struggle in later CIT courses due to a lack of motivation or feeling disconnected from the course content. A similar benefit is seen from efforts to help students identify problems outside of the classroom that will limit their ability to succeed in college. By connecting them with the appropriate support resources, they will hopefully resume their college efforts when they are better positioned for success.

Finally, the IT-specific FYE course allows the department to set the expectation that IT is about more than technical skills. Prompting students to build their study skills, time management practices, reflective practices, and other skills promotes lifelong learning and allows later courses in the curriculum to build on these skills.

This course will continue to develop and evolve, with current efforts focusing on improving the main content around graduation planning and adapting to the changing needs of incoming students. Efforts to assess the course using retention data will continue, and in the meantime, an exit survey is being considered to gather data on student perspectives about the course.

7. REFERENCES

Albarakati, N., DiPippo, L., & Fay-Wolfe, V. (2021). Rethinking CS0 to Improve Performance and Retention. In *Australasian Computing Education Conference* (pp. 131-137).

Becker, K. (2006). How much choice is too much? *ACM SIGCSE Bulletin, 38*(4), 78–82.

Bers, T. & Younger, D. (2014). The First-Year

Experience in Community Colleges. *New Directions for Institutional Research, 160*, 77-93.

Bureau of Labor Statistics. (n.d.). Occupational Outlook Handbook: Computer and Information Technology Occupations. Retrieved May 18, 2022, from https://www.bls.gov/ooh/computer-and-information-technology/home.htm.

Cabo, C., & Satyanarayana, A. (2018). Promoting Students' Social Interactions Results in an Improvement in Performance, Class Attendance and Retention in First Year Computing Courses. In *2018 IEEE Frontiers in Education Conference (FIE)*, 1-8.

Carter, L. (2006). Why students with an apparent aptitude for computer science don't choose to major in computer science. *ACM SIGCSE Bulletin*, 38(1), 27-31.

Connolly, R., Miller, J., Uzoka, F., Lunt, B., Schroeder, M., Miller, C., & Habinka, A. (2016). Red Fish Blue Fish: Reexamining Student Understanding of the Computing Disciplines. *Proceedings of the 17th annual conference on information technology education (pp. 115-120)*.

DeClue, T., Kimball, J., Lu, B., & Cain, J. (2011). Five focused strategies for increasing retention in Computer Science 1. *Journal of Computing Sciences in Colleges, 26*(5), 252-258.

Erickson, S. & Stone, M. (2012). First Year Experience Course: Insights From The First Two Years. *American Journal of Business Education, 5*(2), 139-147.

Guzdial, M. (2019). A Minuscule Percentage of Students Take High School Computer Science in the United States: Access Isn't Enough. *BLOG@CACM,* Retrieved May 18, 2022, from *https://cacm.acm.org/blogs/blog-cacm/239199-a-minuscule-percentage-of-students-take-high-school-computer-science-in-the-united-states-access-isnt-enough/fulltext*

Hanson, M. (2021). "College Dropout Rates" Retrieved May 5, 2022, from https://educationdata.org/college-dropout-rates

Hattie, J., Biggs, J., & Purdie, N. (1996). Effects of Learning Skills Interventions on Student Learning: A Meta-Analysis. *Review of Educational Research, 66*(2), 99-136.

"History – University 101 Program." (n. d.). Retrieved May 4, 2022, from https://www.sc.edu/about/offices_and_divisions/university_101/about/history/index.php

Howard, H. & Jones, W. (2000). Effectiveness of a Freshman Seminar in an Urban University: Measurement of Selected Indicators. *College Student Journal, 34*(4), 509-515.

Humpherys, S., Babb, J., & Abdullat, A. (2015). Where do Student Outcomes Begin? Developing Professional and Personal Management Skills as a Strategy for Student Success in the First Computing Course and Beyond. *Information Systems Education Journal, 13*(6), 4-26.

Kenny, R. W., & Boyer Commission on Educating Undergraduates in the Research University. (1998). Reinventing undergraduate education: A blueprint for America's research universities. New York: The Commission.

Kuh, G. (2008). High-Impact Educational Practices: What they are, who has access to them, and why they matter. *Association of American Colleges and Universities*, Washington, DC.

Lang, D. (2007). The Impact of a First-Year Experience Course on the Academic Performance, Persistence, and Graduation Rates of First-Semester College Students at a Public Research University. *Journal of The First-Year Experience & Students in Transition, 19*(1), 9-25.

Mathis, S. (2008). Introductory Course Improves Retention, Especially For Women. *Information Systems Education Journal, 6*(50), 3-8.

Mayo, T. (2013) First-Year Course Requirements and Retention for Community Colleges. *Community College Journal of Research and Practice, 37*(10), 764-768.

National Academies of Sciences, Engineering, and Medicine. (2018). Assessing and responding to the growth of computer science undergraduate enrollments. National Academies Press.

Ott, L. (2014) Exploration in Computing: Could This Be the Key to Retention? *2014 IEEE Frontiers in Education Conference (FIE) Proceedings, Frontiers in Education Conference (FIE), 2014,* 1-9.

Reese, D. S., Jankun-Kelly, T. J., Henderson, L., & Lee, S. (2013). Impact on retention from a change in undergraduate computing curricula. In 2013 ASEE Southeast Section Conference Proceedings.

Roberson, C. (2011). Aligning generations to improve retention in introductory computing courses. *Journal of Computing Sciences in Colleges, 26*(6), 30-36.

Ryan, M. & Glenn, P. (2004). What Do First-Year Students Need More: Learning Strategies Instruction or Academic Socialization? *Journal of College Reading and Learning, 34*(2), 4-28.

Saulnier, B., Brooks, N., Ceccucci, W., & White, B. (2007). Learning Communities in Information Systems Education: Developing the Reflective Practitioner. *Information Systems Education Journal, 5*(4), 3-8.

Stephenson, C., Derbenwick Miller, A., Alvarado, C., Barker, L., Barr, V., Camp, T., Frieze, C., Lewis, C., Cannon Mindell, E., Limbird, L., Richardson, D., Sahami, M., Villa, E., Walker, H., & Zweben, S. (2018). Retention in Computer Science Undergraduate Programs in the U.S.: Data Challenges and Promising Interventions. New York, NY. ACM.

Editor's Note:

This paper was selected for inclusion in the journal as an EDSIGCON 2022 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 2022.

Appendix A

Learning outcomes for IT-specific FYE course.

1. Understand what a liberal arts education is and its role in their academic experience and success.
2. Demonstrate active participation and engaged learning in class discussions and activities.
3. Integrate their personal, academic, and career goals in relation to their values, interests, and skills.
4. Utilize campus resources and e-tools in pursuit of academic and co-curricular goals.
5. Demonstrate an awareness of the relationship between culture and identity within themselves and others.
6. Examine how their behavior and decisions have an impact on their personal well-being and on their communities.
7. Outline an intentional, integrated plan of curricular and co-curricular learning.
8. Describe the goals and activities of IT professionals across a variety of domains and organizations.
9. Describe the roles and responsibilities of various stakeholders within IT organizations.
10. Identify technological problems and seek solutions collaboratively.
11. Discuss various ways in which to communicate technological information to a variety of stakeholders.

Learning outcomes 1 – 7 are from the university-level UNV 101 FYE course. Learning outcomes 8 – 11 are additional outcomes for the IT-specific FYE course.

Appendix B

List of topics and assignments from a recent section of the IT-specific FYE course.

Week	Topics	Assignments
1	Course Introduction The transition from HS to college What is IT? discussion	Link LMS calendar to Google calendar Develop questions for speakers
2	Mindset • College Mindset • Growth/Fixed Mindset	Mindset reflection Personal vision for college and future
3	Time Management SMART goals	Set a SMART goal Try a new time management practice and reflect on it
4	Critical Thinking Fake News	Spotting Fake News
5	Guest Speaker Professional Networking	Reflection on guest speaker Create LinkedIn profile and make connections Danger of a Single Story (critical thinking)
6	Personal Wellness Taking Care of Others Overview of Jobs in IT	Personal wellness assessment Find an IT job posting. Post in LMS discussion and discuss postings from other students
7	No class - Fall Break	
8	Guest Speaker Introduce graduation planning resources	Reflection on guest speaker Find graduation planning resources for student's specific course of student Online discussion about stress and technology
9	Graduation Planning – led by chief department advisor Academic standing, probation, etc.	Develop graduation plan and meet with assigned advisor SMART goal update
10	Guest speaker Discussion of IT job descriptions	Reflection on guest speaker Online courses with quizzes on financial literacy topics
11	No class	
12	Agile activity	Review and online discussion of BLS data on computing fields
13	Financial Literacy	Complete budget spreadsheet (template provided). Submit reflection on budgeting process.
14	No class - Thanksgiving	
15	In small groups each student is interviewed on their career and education goals.	Submit recorded interviews. Record elevator pitch Final reflection on course.