INFORMATION SYSTEMS EDUCATION JOURNAL

Volume 20, No. 2
April 2022
ISSN: 1545-679X

In this issue:

iscap Information Systems & Computing Academic Professionals

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

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

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

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

INFORMATION SYSTEMS EDUCATION JOURNAL

Exposing the IT Skills Gap: Surveying Employers' Requirements in Four Key Domains

Peter Draus
draus@rmu.edu
Computer and Information Systems Department
Robert Morris University
Moon Township, PA 15108 USA

Sushma Mishra
mishra@rmu.edu
Computer and Information Systems Department
Robert Morris University
Moon Township, PA 15108 USA

Kevin Slonka
slonka@pitt.edu
Information Sciences Department
University of Pittsburgh
Greensburg, PA 15601 USA

Natalya Bromall
bromall@rmu.edu
Computer and Information Systems Department
Robert Morris University
Moon Township, PA 15108 USA

Abstract

Information Technology (IT) skills gap discourse suggests a mismatch between what students are acquiring in terms of knowledge and skills in their education versus what employers believe are useful skills for doing day to day tasks. This study builds upon previous research (analyzing the skills of college students in IT-related majors) by surveying industry professionals to determine the skills their organization requires and offering suggestions that can benefit the educational institutions and create a better educated workforce. Implications are drawn and a conclusion is presented.

Keywords: IT skills, competencies, cyber security, infrastructure, development, emerging technologies

1. INTRODUCTION

Year after year, the demand for Information Technology (IT) specialists grows. The expected growth rate from 2019 to 2029 is estimated to be 11%. "IT specialist," however, isn't a single job type; it's a broad category. This adds underlying levels of complexity to that projection. The U.S.

Bureau of Labor Statistics (BLS) groups these jobs into the "Computer and Information Technology" category and further splits it into ten sub-categories. Although being different nomenclature from the above grouping, it is in the eliciting of specific job titles from those sub-categories where the complexity emerges (U.S. Bureau of Labor Statistics, 2020).

The BLS uses the nomenclature of Computer Programmer or Software Developer, but that can manifest in industry job postings as "Full Stack Software Engineer" or "Quality Assurance Engineer." A decade or longer ago, this would not have been the case; job postings would have used the simple labels denoted by the BLS. Industry, however, no longer operates using these simple labels. The change in nomenclature points to a change in expectations of the industry. Whereas IT-related jobs previously operated in silos, we see a shift to more collaborative structures. Someone searching for a job used to use the terms "software developer," where the expectation would be that they only need skills relating to software development. If one were to do a job search today, they would find listings for Full Stack Software Engineers that require skills spanning multiple categories, whereas the BLS categorized Computer Programmers, Database Administrators, Software Developers, and Web Developers as four different jobs. To be successful in today's industry, the Full Stack Software Engineer must have all of those skills. This is the case in most areas of IT. The norm is now integrated, broad, and collaborative knowledge as opposed to individual silos.

Dawson & Thomson (2018) furthered this discussion by examining the need for skills outside the technical realm. While their study pertained to the cybersecurity workforce, this can be generalized to the entirety of IT. They found that domain-specific knowledge and social intelligence were key categories of skills that reached beyond the technical. In order to be successful, employees must have such vital traits as being a systemic thinker, being a team player, having technical and social skills, being loyal to the organization, having strong communication skills, and being a continual learner. Others like Huang, Kvasny, Joshi, Trauth, & Mahar (2009) examined the shift in demands of the industry and noted that skills that did not exist previously have been introduced, such as data warehousing.

The entry-level education requirements of the industry are also changing. 70% of IT-related employers specifically require a Bachelor's degree as a minimum (Robin, 2011). The BLS corroborates this finding, listing the entry-level education requirement as a Bachelor's degree or higher for IT jobs in eight of the 10 categories. Recently, attempts to alter these education requirements have been trending. The push is for more focus being given to certifications and work experience than to college degrees (DevMountain, 2021; Dietrich, 2018; Indeed, 2020). As previously explained, the reality is that well-rounded employees that are molded by a college education are still preferred by companies over those employees with just technical skills.

It is difficult for students to make sense of this skill landscape given the complexities and discrepancies. Furthermore, there is no evidence that students' perception of necessary industry skills aligns with the skills actually required by industry. This study serves to build upon a previous study on student perceptions of skills (Slonka, Bromall, Mishra, & Draus, 2021) by analyzing data from industry organizations and comparing it to the student data. Suggestions will be made that can positively affect both the students and the organizations by arriving at a clearer view of the skills gap.

The majority of the research questions answered by this study mirror those of the previous study except that this study's questions ascertain the viewpoint of industry professionals, not students. The full list of research questions is the following:

RQ1: What levels of experience in the Cyber Security domain do employers expect from IT graduates?

RQ2: What levels of experience in the Infrastructure & Operations domain do employers expect from IT graduates?

RQ3: What levels of experience in the Software Development domain do employers expect from IT graduates?

RQ4: What levels of experience in the New & Emerging Technologies domain do employers expect from IT graduates?

RQ5: To what degree do employers expect experience in non-primary hiring domains?

RQ6: To what degree do employer's knowledge expectations differ from student's self-reported knowledge?

The remainder of the paper is organized as the following. The introduction section above is followed by a critical review of the relevant literature in the field. Literature review is followed by data collection and analysis section. The results are presented. The discussion section draws the implications of our results followed by the conclusion of the study.

2. REVIEW OF LITERATURE

As one of the most dynamic and rapidly changing fields, Information Technology has a large gap between the knowledge and experience required by the employers, and the ones received by the recent graduates in their academic programs. After graduation the students have to be prepared to a rigorous training, and even the experienced IT specialists are often required to update or change their skills (Koh, Lee, Yen, & Havelka, 2004). Many researchers note a gap between the knowledge perceived necessary by the IT faculty and the knowledge searched by the organizations (Aasheim, Li, & Williams, 2009). Taylor-Smith, Berg, Smith, Meharg, Fabian, & Varey (2019) note that this gap is two-fold. First, there is an overall deficit in digitally skilled workers. Second, the current IT employees and recent graduates demonstrate the lack of appropriate skills. Taylor-Smith et al. also mention that the employees currently prefer that the new hires do not require a substantial training and tend to spend less resources on training.

After a decline in IT programs enrollment observed 15 years ago (Abraham, Beath, Bullen, Gallagher, Goles, Kaiser, & Simon, 2006), the faculty faced a daunting task to create a better match between the requirements of the job market and the content of college programs. In addition to the obvious reason of being more successful in the job market, the students whose skills better match the job market have an increased interest in the content of their programs, and there are multiple studies supporting that. For example, Kapoor & McCune (2018) demonstrated that if students make a clear connection between the knowledge they receive and the real-world applications, they become more successful as professionals. One possibility to improve the students' qualifications before they enter the job market is to give them a professional field training such as internships; however, such training and experiences are still rare (McKenzie & Coldwell-Nielson, 2018).

In this paper we divide the IT skills into four very general categories: Infrastructure & Operations, Software Development, Cyber Security, and Emerging Technologies; however, we anticipate that many professionals are required to be proficient in more than one area. In fact, in spite of an existing public opinion that the IT job requirement has become more "narrow" and broad knowledge is no longer needed, many research studies disprove this statement. Hollister, Spears, Mardis, Lee, McClure, & Liebman (2017) interviewed IT recruiters and found out that the recent college graduates are expected to have knowledge in a broader set of disciplines (database management, programming, security, networking, soft skills, etc.), but this knowledge is not expected to be in-depth.

Many researchers came to the conclusion that, in addition to professional skills, the successful job candidates need excellent soft skills. For example, Haney and Lutters (2016) found that the candidates interviewed for the Cyber Security jobs were expected to possess the innovative skills and the skills to address social and organizational issues, in addition to their professional qualifications. In their study, Dawson & Thompson (2018) concluded that the expectations from a successful candidate are to be a systemic thinker, a team player, have strong communication skills and prepare for continuous learning. With a variety of different skill requirements for the Cyber Security professionals, the standards for the corresponding programs are also set at a very high level. Faculty, courses and the supporting infrastructure such as labs and equipment are listed as the most essential parts of the education process (Dampier, 2015). According to Purdue Global (2018), the Cyber Security graduates must possess the knowledge across six disciplines: Network Security, Digital Forensics, Cybersecurity Policies, Cybersecurity Ethics and Law and Information Systems Security. Although Cyber Security remains an area of IT with a great demand in workforce, it is not the only one on the jobs list. According to Hollister et al. (2017), Infrastructure/Architecture and Operation Support are among the most desirable skill areas, while the CompTIA report of 2021 includes a list of emerging technologies that are in great demand, such as Machine Learning and Artificial Intelligence, Data Analytics and Big Data, and Cloud Technologies (Madden, 2021).

3. METHODS AND RESULTS

Data Collection

An online survey was developed to ascertain hiring professionals' within the IT field self-defined expectations for new hires on their level of knowledge in four domains in the IT field. In addition to the demographic questions, the survey instrument asked the participant to indicate the level of expertise required for positions in each of the sub-domains, which resulted in 44 questions. These questions elicited the needed experience on a 5-point Likert scale (1=No experience is needed in this area, 2=Basic experience: the applicant should understand the concept, 3=Some experience, 4=Substantial experience: the applicant should be ready to work on a team, 5=Highest level of experience: the applicant should be ready to work alone). Participants were also asked in which of the four domains they *primarily* hire employees and in which of the four domains they ever hire employees (allowing more than one selection).

84 subjects from Western PA IT associations completed the survey. The data from the subject pool showed subjects with many years of experience and a high-level job position. Of those, 37 reported their current job title. 15 were at the C level (CEO/CIO/CTO, president…), 10 at the manager level, and 12 reported being at the Senior level of their role in the organization. Of those reporting their years of experience (N=48), 56% had greater than 15 years with only 8 percent having less than 5 years' experience. On the education side, 65% reported having a bachelor's degree and 30% having earned a master's degree. Only one subject reported having earned a doctorate.

The subjects selected from a list of four areas (Cyber security, Infrastructure & Operations, Development, or New Technologies) as their primary hiring area. As could be expected none of the subjects reported hiring in the "New technologies" area. The majority (62%) reported Cyber Security as their main hiring areas with Infrastructure and operations (25%) and Development (13%) making up the rest. When asked which of the four domains are they actually hiring in the result were Cyber Security (43%), Infrastructure and Operations (26%) Development (17%) and New and Emerging Technologies (21%).

Results

RQ1: What levels of experience in the Cyber Security domain do employers expect from IT graduates?

The overall results for the Cyber Security domain are shown in Appendix A Table 1. Mean values in all domains were at the 'Some' level except for "Security Architecture and Engineering" with a mean (3.85) in the Substantial range. Only one domain had any subjects select at the "None" level. which was in Security Operations and only one domain did not have any ranking at "Highest" which was Asset Security.

The results are consistent with the expectations from fresh hires in the field of cyber security. New graduates are mostly employed in architecture implementation jobs, such as configuration and management of policies, devices, development issues, and risk management.

RQ2: What levels of experience in the Infrastructure & Operations domain do employers expect from IT graduates?

As can be seen in Appendix A Table 2, the employers' highest mean score (3.44) on "Cloud Administration and Support" joined all of the other skills in the "Some Experience" range. Cloud administration and support is quite prevalent and graduates are expected to have working knowledge in AWS or Azure environments to successfully transition in -ground realities of organizational IT functions.

RQ3: What levels of experience in the Software Development domain do employers expect from IT graduates?

All of the skills in this domain were rated in the "Some" level of experience with "Programming Logic" having the highest mean at 3.42 and "Abstraction" (2.81) having the lowest mean rating. All of the results are shown in Appendix A Table 3. Employers understand the changing needs of programming languages and its ability to adapt to newer technologies at a fast pace. It is pertinent that graduates have a solid understanding of topics that constitute fundamentals of programming, such as programming logic, configuring the environment, and testing and debugging. These skills allow graduates to adapt swiftly to newer ways of doing things.

RQ4: What levels of experience in the New & Emerging Technologies domain do employers expect from IT graduates?

All of the sub-domains were ranked at the "Some" level, but interestingly, the "Cloud Computing and Cloud Technologies" topic had nobody select the "None" option. This may signify that Cloud Computing is no longer considered a new and emerging technology. Cloud computing is easily a more prevalent way of providing IT services within and outside the organization. All of the results are shown in Appendix A Table 4.

RQ5: To what degree do employers expect experience in non-primary hiring domains?

Appendix A Table 5 shows the means for each of the four domains grouped by the self-reported primary area of hiring for each subject. As can be seen in Appendix A Table 5, there were no subjects who selected New and Emerging Technologies as their primary hiring area. As can be expected, the highest rated domain for each area was the corresponding area to the subject's hiring domain. It still should be noted that the means in all areas, both primary and non-primary, were all within the same "Some" level of experience. "New and Emerging Technologies" is not clearly defined and refers to all other domains that are not listed as a primary domain. This lack of clarity about the domain results in every respondent interpreting his or her own meaning of the domain. It is not surprising that emerging technologies was not high on the priority list of employers; organizations hire graduates to perform well-defined established tasks.

RQ6: To what degree do employer's knowledge expectations differ from student's self-reported knowledge?

To calculate the difference between the employer expectations and students' self-reported skill levels, the mean of the students was subtracted from the mean of the employers. Appendix A Table 6 shows the sub-domains with differences higher than 1 and Appendix A Table 7 shows sub-domains with differences lower than 1.

The two lowest differences lay in the Infrastructure and Operations Domain: "Application Installing, Configuration and Deployment" and "Desktop Support" and could be called equal.

Appendix A Tables 6 and 7 combined show means across all domains for both students and employers. For the employers, all but two are at the "Some" level of experience. Security Architecture and Engineering had the highest mean (3.85), which is at the "Substantial" level of expected experience. Crypto Currency had the lowest overall Mean (2.46), which was the only one at the "Basic" level. Again, it should be noted the high mean (3.27) for the Cloud Computing and Cloud Technologies ranked at position 7 on the list. It probably should not still be listed as a new and emerging technology.

As can be seen in Table 1, all four domains are present in both tables. Only the New and Emerging Technologies sub-domain shows any real difference in distribution between the higher differences and lower differences tables. This clearly shows that the employers are expecting a higher level of knowledge in these technologies than the students are planning to obtain. IT skills and knowledge are robust in nature and students need to understand that to succeed in this field it requires continuous education and learning.

Domain	**Difference > 1**		**Difference < 1**	
	N	**%**	**N**	**%**
Security	4	17%	3	17%
Inf/Op	6	26%	7	39%
Develop	5	22%	4	22%
New	8	35%	3	17%

Table 1: Distribution of Domains

Further analysis between the results of this study (employer perceptions) and the results of the antecedent study (student perceptions) point toward the same conclusion. If the differences presented in Appendix A Table 6 are sorted by the student score one will find that the measure of the student score correlates with the severity of the difference, shown in Table 2.

Domain: Sub-Domain	**+/-**
Security: Security Architecture and Engineering	-2.04
Inf/Op: Cloud Administration and Support	-1.54
Develop: Container Application	-1.53
Security: Software Development Security	-1.46
Inf/Op: SLA: SL Objective and SLI	-1.44
New: Enterprise/Intelligent Automation	-1.39
New: Robotics Process Automation	-1.38
Develop: Version Control/Deploy/Config/Environment	-1.37
New: Machine Learning and AI/Analytics	-1.31
New: Edge Computing	-1.31

Table 2: Gaps Greater than 1.30

4. DISCUSSION

The IT skills gap suggests a disconnect with what employers want from a graduate and what students are learning in academic institutions. Our study implies that employers expect a certain level of proficiency in IT areas from its hires including new and emerging technologies. The employers are looking for knowledge in current IT domains and competency in fundamental concepts that allow graduates to constantly learn new technologies, adapt, and flourish.

As shown in Table 2, the major gaps are due to the student views being out of line with what is expected by employers. One may infer these gaps are as much about what is *not* being taught in schools as opposed to *how* it is being taught, or even about the amount of schooling the students have obtained. It is important for academia and industry to collaborate closely and for these partnerships to provide opportunities to train the "job ready" graduates in a way that is beneficial to students and employers. The IT industry needs comprehensive academic, technical, and professional competencies and knowledge, skills, and abilities (KSAs) that may not be adequately addressed by traditional college classroom activities. Collaborations and partnerships between information technology (IT) education providers, programs, and industry organizations to improve education and serve the needs of the industry are important (Wang et al, 2020).

This study has implications for practitioners. The study provides a repertoire of topics, skills, and knowledge areas that are typically taught at the university level to IT graduates. Employers can look at the spectrum of content area and identify ways to use the training provided to students for their own needs. This allows organizations to fully utilize the knowledge and competency of hires in a systematic way. This study provides insights into what industry demands of fresh college graduates and university administrators need to gauge the expectations of future employers and adapt in a realistic manner. Additionally, as more and more companies offer on-the-job training or apprenticeship programs (Dishman, 2017; O'Donnell, 2021) it will be important for employers and academic agencies to work closely for mutual benefits and to ensure that this synergistic cohabitation will have long-term implications for all involved parties.

This study has implications for research as well. The first implication is that this study identifies an urgent need to develop collaborative mechanisms for the exchange of information between academia and industry in a meaningful way. Second, more studies are required to refine the expectations of employers in each IT domain such that a more clear and precise understanding of knowledge, skills, and competency is developed. This clarity will allow embedding all the topics in a holistic way.

Although the response rate and participant pool's narrow geographical location could be seen as a limitation, as statistical studies benefit from large response rates, this study does not find those factors limiting. The geographic area from which the participants were pooled represent a microcosm of the nation's technical workforce, with companies ranging from small businesses with less than 10 employees to large corporations, such as Amazon and Google. Additionally, the self-reporting bias plays a role in all research such as this. Because this is such a critical topic in academia and industry, future research should be undertaken to expand the scope.

5. CONCLUSION

This study surveyed IT employers about their expected level of expertise from recent graduates from information systems/technology programs. They survey was based on four specific IT domains: cyber security, infrastructure and operations, software and development, and new and emerging technologies. The results indicated that employers expect proficiency in fundamental topics such as programming logic and debugging and a basic understanding of advanced topics. However, the employers also showed a high preference for skills and competency in the new and emerging technology domain. The results were explained and implications were drawn; the main implication being more collaborative partnership with academia and industry for mutual benefits.

6. REFERENCES

Aasheim, C.L., Li, L. & Williams, S. (2006). Knowledge and skill requirements for entry-level information technology workers: a comparison of industry and academia. *Journal of Information Systems Education, 20*(3), pp. 349-356.

Abraham, T., Beath, C., Bullen, C., Gallagher, K., Goles, T., Kaiser, K., & Simon, J. (2006). IT Workforce Trends: Implications for IS Programs. *Communications of the Association for Information Systems, 17*, pp. 1147-1170.

CC2020, (2020). *Computing Curricula 2020 Paradigms for Global Computing Education*. https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf

Dampier, D. (2015). Building a successful cyber-security program. *Distributed Analytics and Security Institute, Mississippi State University.* http://www.dasi.msstate.edu/publications/docs/2015/06/13502Cyber_Security_Workshop_paper_-_Final.pdf

Dawson, J. & Thomson, R. (2018). The future cybersecurity workforce: Going beyond technical skills for successful cyber performance. *Frontiers in Psychology*, *9*, p. 12.

DevMountain. (2021, March 18). *Certifiable? Finding programmer jobs without a degree.* https://blog.devmountain.com/certifiable-finding-programmer-jobs-without-a-degree/

Dietrich, E. (2018, April 9). *How to get a programming job without a degree*. https://daedtech.com/programming-job-without-degree/

Dishman, L. (2017). *These Top Tech Companies Are Hiring First, Training Later*. https://www.fastcompany.com/40482650/these-top-tech-companies-are-hiring-first-training-later

Haney, J., M., & Lutters, W.G., (2016). *Skills and characteristics of successful cybersecurity advocates.* https://www.usenix.org/system/files/conference/soups2017/wsiw2017-haney.pdf

Hollister, J.M., Spears, L.I., Mardis, M.A., Lee, J., McClure, C.R., & Liebman, E. (2017). Employers’ perspectives on new information technology technicians’ employability in North Florida. Education and Training. *London, 59*(9), pp. 929-945.

Huang, H., Kvasny, L., Joshi, KD, Trauth, E., & Mahar, J. (2009, May). *Synthesizing IT job skills identified in academic studies, practitioner publications and job ads* [Conference paper]. Proceedings of the special interest group on management information system's 47th annual conference on Computer personnel research, SIGMIS-CPR, Limerick, Ireland, 121-128.

Indeed Career Guide. (2020, December 22). *10 high-paying IT jobs you can get without a degree*. https://www.indeed.com/career-advice/finding-a-job/it-jobs-without-degree

Kalra, S., Thevathan, C., & Hamilton, M. (2020). *Developing Industry-Relevant Higher Order Thinking Skills in Computing Students.* Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE ’20), June 15–19, 2020, Trondheim, Norway. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3341525.3387381

Kapoor, A. & Gardner-McCune, C. (2018). *Understanding Professional Identities and Goals of Computer Science Undergraduate Students.* Proceedings of SIGCSE’18, February 21-24, 2018, Baltimore, MD, USA.

Koh, S., Lee, S., Yen, D.C. & Havelka, D. (2004). The relationship between information technology professionals' skill requirements and career stage in the e-commerce era: an empirical study. *Journal of Global Information Management; Hershey, 12*(1), pp. 68-82.

Madden, J. (2021). *Top IT skills in demand in 2021.* https://www.comptia.org/blog/top-it-skills-in-demand

McKenzie, S. & Coldwell-Nielson, J. (2018). Understanding the career development and employability of information technology students. *Journal of Applied Research in Business Education; Bingley, 10*(4), pp. 456-468.

O’Donnell, B. (2021). *Looking to level up? Amazon, Google, Microsoft and more offer training programs.* https://www.usatoday.com/story/tech/columnist/2021/04/26/amazon-google-and-more-offer-training-programs-newcomers/7335646002/

Pham, A. & Dao, H. (2020). *The Importance of Soft Skills for University Students in the 21st Century.* Proceedings of ICAAI 2020, October 09–11, 2020, London, United Kingdom, Association for Computing Machinery.

Purdue Global (2018). *Bachelor of Science in Cybersecurity.* https://www.purdueglobal.edu/degree-programs/information-technology/bachelors-cybersecurity.pdf

Robin, G. J. (2011, May). *Do companies look for education, certifications, or experience: a quantitative analysis* [Conference paper]. Proceedings of the 49th SIGMIS annual conference on computer personnel research, SIGMIS-CPR, San Antonio, TX, USA, 1-5.

Slonka, K., Bromall, N., Mishra, S., & Draus, P. (2021). IT skills gap: A survey of IT students' knowledge in 4 key domains. *Issues in Information Systems, 22*(2), 175-184.

Song, X., Huang, X., & Huang, K. (2019). *Research on the Effect of Skill Mismatch on Skill Development and Job Satisfaction* *among Graduates.* Proceedings of ICEME 2019, July 15–17, 2019, Beijing, Association for Computing Machinery.

Taylor-Smith, E., Berg, T., Smith, S., Meharg, D., Fabian, K. & Varey, A. (2019). *Bridging the Digital Skills Gap.* Proceedings of ITiCSE '19, July 15–17, 2019, Aberdeen, Scotland UK, Association for Computing Machinery.

U.S. Bureau of Labor Statistics. (2020, September 1). *Computer and Information Technology Occupations*. https://www.bls.gov/ooh/computer-and-information-technology/home.htm

Van Slyke, C., Clary, G., Ellis, S. & Maasberg, M. (2019). *Employer Preferences for Cybersecurity Skills among Information Systems Graduates.* Proceedings of the SIGMIS-CPR '19, June 20–22, 2019, Nashville, TN.

Wang, P., Hayes, N., Bertocci, M., Williams, K., & Sbeit, R. (2020). *The Role of Industry Partnerships and Collaborations in Information Technology Education.* 17th International Conference on Information Technology–New Generations (ITNG 2020) (pp. 9-15). Springer, Cham.

Appendix A

	Level of Experience %					
Sub-Domains	**None**	**Basic**	**Some**	**Substantial**	**Highest**	**Mean**
Security Architecture and Engineering		20.8	14.6	50	14.6	3.85
Communications and Network Security		16.7	35.4	41.7	6.3	3.38
Software Development Security		25	22.9	43.8	8.3	3.35
Security and risk management		18.8	39.6	39.6	2.1	3.25
Security Assessment and Testing		25	39.6	29.2	6.3	3.17
Security Operations	2.1	33.3	25	35.4	4.2	3.06
Identity and Access Management		29.2	39.6	29.2	2.1	3.04
Asset Security		33.3	47.9	18.8		2.85

Table 1: Cyber Security

	Level of Experience %					
Sub-Domains	**None**	**Basic**	**Some**	**Substantial**	**Highest**	**Mean**
Cloud Administration and Support	2.1	16.7	29.2	39.6	12.5	3.44
Production Environment Support		20.8	37.5	35.4	6.3	3.27
Incident Management	2.1	22.9	33.3	31.3	10.4	3.25
Automation – Scripting		25	35.4	29.2	10.4	3.25
Operating Systems	4.2	18.8	35.4	33.3	8.3	3.23
Networking		16.7	47.9	33.3	2.1	3.21
Server Management		25	43.8	22.9	8.3	3.15
Monitoring, Alerting, Notification	2.1	31.3	33.3	22.9	10.4	3.08
App Installing, Config and Deployment	6.3	29.2	31.3	22.9	10.4	3.02
SLA: SL Objective and SLI		33.3	39.6	22.9	4.2	2.98
Testing Environment Support	4.2	25	47.9	14.6	8.3	2.98
Storage	4.2	31.3	41.7	16.7	6.3	2.9
Desktop Support	12.5	31.3	35.4	12.5	8.3	2.73

Table 2: Infrastructure & Operations

	Level of Experience %					
Sub-Domains	**None**	**Basic**	**Some**	**Substantial**	**Highest**	**Mean**
Programming Logic	2.1	14.6	35.4	35.4	12.5	3.42
Version Control/Deployment/Configure /Environment	4.2	14.6	37.6	39.6	4.2	3.25
Testing and Debugging	4.2	12.5	50	25	8.3	3.21
Data Modeling	4.2	16.7	35.4	43.8		3.19
Database administration, management and development	2.1	20.8	39.6	35.4	2.1	3.15
Container Application	4.2	16.7	47.9	25	6.3	3.12
Event Handling/Interrupts	4.2	25	43.8	25	2.1	2.96
User Interface Design/HCI	12.5	20.8	31.3	31.3	4.2	2.94
Abstraction	6.3	33.3	35.4	22.9	2.1	2.81

Table 3: Development

	Level of Experience %					
Sub-Domains	**None**	**Basic**	**Some**	**Substantial**	**Highest**	**Mean**
Cloud Computing and Cloud Tech		16.7	45.8	31.3	6.3	3.27
Data Science	4.2	25	33.3	31.3	6.3	3.1
Machine Learning and AI/Analytics	8.3	20.8	37.5	25	8.3	3.04
Enterprise Automation/Intelligent Automation	4.2	33.3	27.1	31.3	4.2	2.98
Internet of Things	6.3	27.1	35.4	29.2	2.1	2.94
Edge Computing	4.2	33.3	35.4	25	2.1	2.88
Robotics Process Automation	16.7	22.9	29.2	25	6.3	2.81
Block Chain	12.5	29.2	31.3	22.9	4.2	2.77
Quantum Computing	25	27.1	16.7	22.9	8.3	2.62
Virtual Reality and Augmented Reality	23.4	29.8	25.5	17	4.3	2.49
Crypto Currency	27.1	25	25	20.8	2.1	2.46

Table 4: New & Emerging Technologies

Primary Area	**N**	**Means in the Domains**			
		CS	**I/O**	**Dev**	**New**
Cyber Security	29	3.15	2.95	3.03	2.88
Infrastructure and Operations	12	3.69	3.71	3.5	3.03
Development	6	2.66	2.83	3.1	2.87
New and Emerging Technologies	0				
Overall	47	3.23	3.13	3.13	2.87

Table 5: Primary Domain v Means in each Domain

Sub-Domains	**Domain**	**Employers**	**Students**	**Difference**
Security Architecture and Engineering	Security	3.85	1.81	2.04
Cloud Administration and Support	Inf/Op	3.44	1.9	1.54
Container Application	Develop	3.12	1.59	1.53
Software Development Security	Security	3.35	1.89	1.46
SLA: SL Objective and SLI	Inf/Op	2.98	1.54	1.44
Enterprise/Intelligent Automation	New	2.98	1.59	1.39
Robotics Process Automation	New	2.81	1.43	1.38
Version Control/Deploy/Config/Environment	Develop	3.25	1.88	1.37
Machine Learning and AI/Analytics	New	3.04	1.73	1.31
Edge Computing	New	2.88	1.57	1.31
Data Modeling	Develop	3.19	1.91	1.28
Automation - Scripting	Inf/Op	3.25	1.98	1.27
Production Environment Support	Inf/Op	3.27	2.01	1.26
Block Chain	New	2.77	1.54	1.23
Abstraction	Develop	2.81	1.6	1.21
Quantum Computing	New	2.62	1.41	1.21
Cloud Computing and Cloud Technologies	New	3.27	2.09	1.18
Incident Management	Inf/Op	3.25	2.08	1.17
Event Handling/Interrupts	Develop	2.96	1.79	1.17
Data Science	New	3.1	1.99	1.11
Security Assessment and Testing	Security	3.17	2.13	1.04
Identity and Access Management	Security	3.04	2.03	1.01
Testing Environment Support	Inf/Op	2.98	1.97	1.01

Table 6: Sub-Domain Differences Greater Than 1

Sub-Domains	**Domain**	**Employers**	**Students**	**Difference**
User Interface Design/HCI	Develop	2.94	1.97	0.97
Server Management	Inf/Op	3.15	2.2	0.95
Communications and Network Security	Security	3.38	2.44	0.94
Programming Logic	Develop	3.42	2.51	0.91
Security Operations	Security	3.06	2.17	0.89
Security and risk management	Security	3.25	2.37	0.88
Database Admin, Management, & Dev	Develop	3.15	2.28	0.87
Asset Security	Security	2.85	1.98	0.87
Monitoring, Alerting, Notification	Inf/Op	3.08	2.37	0.71
Testing and Debugging	Develop	3.21	2.51	0.7
Networking	Inf/Op	3.21	2.72	0.49
Operating Systems	Inf/Op	3.23	2.88	0.35
Internet of Things	New	2.94	2.59	0.35
Storage	Inf/Op	2.9	2.56	0.34
Virtual Reality and Augmented Reality	New	2.49	2.26	0.23
Crypto Currency	New	2.46	2.25	0.21
App Installing, Config, and Deployment	Inf/Op	3.02	3.01	0.01
Desktop Support	Inf/Op	2.73	2.72	0.01

Table 7: Sub-Domain Differences Less Than 1

A Bot Assisted Instructional Framework for Teaching Introductory Programming Course(s)

Deepak Dawar
daward@miamioh.edu
Dept. of Computer and Information Technology
Miami University
Hamilton, Ohio 45011, U.S.A.

Abstract

Learning computer programming is a challenging task for most beginners. Demotivation and learned helplessness are pretty common. A novel instructional technique that leverages the value-expectancy motivational model of student learning was conceptualized by the author to counter the lack of motivation in the introductory class. The result was a frequency adherent scaffolded instructional technique called An Assignment A Day (AAAD). Instead of writing an assignment and a lab for each module/chapter, students were asked to complete one assignment a day, not exceeding four assignments a week. The assignments were incrementally difficult and had to be done almost every day. With the application of AAAD for two consecutive semesters, there was a meaningful improvement in the final grades. This technique, though initially encouraging, created a significant load on the instructor in terms of assignments graded and questions answered every day. A natural language processing (NLP) based conversational agent was designed and integrated with AAAD to counter this overload. The idea was simple – relay commonly asked course questions to an NLP based chatbot and let the instructor handle the complex queries. This integrated system was named Conversational Agent Supported Scaffolded Approach (CASSA). The main contribution of this work is the construction of a conversational agent and its integration with AAAD. The conversational agent is currently being assessed for overall efficacy, though preliminary results are discussed. The vision is to create a generic virtual assistant template that can be re-used across multiple courses to assist instructors.

Keywords: Conversational agents, NLP, introductory programming, pedagogy, value-expectation, student procrastination.

1. INTRODUCTION

Computer programming is an arduous learning process for most beginners, and high failure rates have been reported continuously (Allan & Kolesar, 1997; Newman, Gatward, & Poppleton, 1970; Bennedsen & Caspersen, 2007; Sheard & Hagan, 1998; Watson & Li, 2014; Beaubouef & Mason, 2005; Howles, 2009; Kinnunen & Malmi 2006; Mendes et al., 2012). Given the complex nature of the programming (Kim & Lerch, 1997; Rogalski & Samurçay, 1990; Robins, Rountree & Rountree, 2003), students frequently get demotivated. While teaching multiple introductory programming courses over many years, the author observed that apart from the complex nature of programming, there were other factors at play that feed the demotivation loop. Some examples are:

- Less than desirable instructor presence
- High temporal disengagement with the programming activities
- Students internal lack of motivation

Keeping these factors in mind, and inspired by value-expectancy (Keller, 1983) & cognitive load theory (Paas, Renkl, & Brünken, 2010; Sweller, 1988, 1994), a novel instructional technique called An Assignment A Day (AAAD) approach was designed. Instead of completing a lab and assignment per chapter, students were asked to complete one simple assignment a day, with a cap of four assignments a week. Every subsequent assignment of a chapter/course built on the previous assignment and carried an incremental cognitive load (see Appendix A). Apart from testing students on new concepts, the subsequent assignment reused the concepts learned/applied in the previous assignment. The approach (Dawar, 2021) can be summarized as:

1. Students will ideally do one assignment per day.
2. Opening assignments of the chapter will test students on very basic skills like writing a method stub. Subsequent assignments will gradually increase in complexity keeping in mind the cognitive load asserted by the assignment. This mechanism is in part based on the study conducted by Alexandron et al. (2014).
3. There will not be more than four assignments per week. Deadlines may be relaxed on a case-to-case basis.
4. As an exception, and depending upon the cognitive load, an assignment may be completed in two or more days rather than a single day.

The technique rests on three central pillars, as shown in Figure 1.

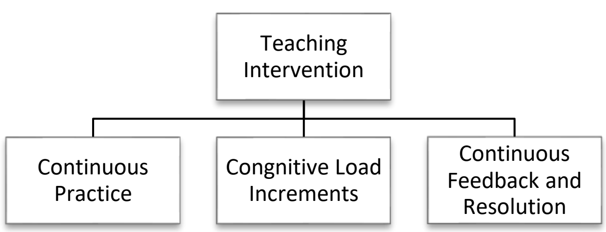


Figure 1: AAAD Interventional Technique

This study aims to address two research questions:

a) What is the effect of mandatory continuous engagement with cognitively germane testing on student outcome and instructor load?
b) How can instructor load be minimized while maintaining the sanctity of the technique?

The author could foresee at least two significant issues that could derail the potential acceptability of this technique:

a) Will the high number of assignments, albeit of germane cognitive load, dissuade students from participating, thereby compounding the very problem the author is trying to tackle, i.e., lack of motivation due to learned helplessness? Constant testing has been associated with high student anxiety (Kaplan et al., 2005). An easy way to make students dislike programming is to put them under unnecessary stress (Goold & Rimmer, 2000). Strict enforcement of everyday deadlines may easily overwhelm these students. The only chance of overcoming this hurdle was providing germane load assignments.
b) Even if the intervention shows promising results with students, what does that mean for the instructor load? More assignments would naturally elicit more questions, requiring additional instructional and tutoring presence, and more grading time, besides other externalities. Massive overload and instructor fatigue become apparent. Some follow-up questions are warranted. For example:
   1. Is it prudent or even feasible to run a potentially beneficial instructional intervention while risking instructor overload simultaneously?
   2. If the intervention is proven to be beneficial, how can instructor support be increased so that the outcome is better for students (in terms of motivation) as well as the instructor (in terms of course load)?
   3. Do the system and tools required for instructor support already exist, or would they need to be allocated/constructed?
   4. Are these support systems course-specific, or can they be reused within courses?

These questions are vast and may need multiple solutions at multiple levels. As a preliminary solution, a conversational agent or a chatbot is proposed to assist the instructor. The essential function of this agent is to answer repeatedly asked student questions in the course when access to the instructor is not available.

The rest of the paper is structured as follows. Section 2 discusses the perceived need for the intervention and the conversational agent and builds a case for their integration. Section 3 touches upon the operational aspects of natural language processor systems (NLP) and illustrates the parts of the conversational agent. Section 4 discusses the preliminary results for the accuracy of the conversational agent.
Section 5 concludes the paper and briefly presents the foundations of future research.

2. A Case for Integration of a Conversational Agent With Scaffolded Instructional System

In this section, justification for building and employing the AAAD technique is presented. It is then argued that while this might be a good idea for student motivation and performance, it can overload the instructor who lacks access to dedicated resources like graders and tutors. A case is then built for the construction and use of a conversational agent/chatbot to take some load off the instructor while not jeopardizing the instructional technique. The terms conversational agent and chatbot are used interchangeably throughout the paper.

A Case for AAAD Approach

Students' belief in their success is vital if they are to be motivated to learn. There are many causes of student demotivation, but the one suspect that the author can categorically point towards in their classrooms is high cognitive load. Cognitive load theory (Paas, Renkl, & Brünken, 2010; Sweller, 1988, 1994) throws light on the aspects of load placed on working memory while a task is being executed. Computer programming requires balancing numerous interactive tasks simultaneously. For example, it involves juggling numerous details like problem domain, the current state of the program, language syntax, strategies (Winslow, 1996).

Procrastination is extremely prevalent in students studying in a university setup. Some estimates suggest that 80 to 95 percent of students engage in procrastination (Steel, 2007). The longer the students wait to turn in the assignment, the worse their grades become (Kim & Seo, 2015). Procrastination has also been linked to higher levels of anxiety, stress, and fatigue (Beutel et al., 2016). After having taught multiple programming courses over multiple years, the author encountered similar patterns.

AAAD was designed keeping these factors in mind. The intervention made continuous targeted interaction between the material and students – somewhat mandatory. It was opined that this would:

- Establish a clear study pattern for students to counter procrastination.
- Potentially improve student's expectations owing to germane cognitive loads.
- Make them practice programming every almost every day. The inspiration for this operation came from strong evidence suggested by psychological studies (Brown & Bennett, 2002; Glover, Ronning & Bruning, 1990; Moors & De Houwer, 2006) done on variable student populations. Constant practice can improve student motivation and make them want to learn more (Moss & Case, 2001).

The technique AAAD was administered to two experimental groups (E1 and E2), and the study was spread over three semesters. The control group (C1) was asked to complete one assignment and one lab work per week. Quizzes were given at the end of every chapter. This is the usual approach followed at our institution for introductory programming classes. E1 and E2 were taught with the interventional approach for the subsequent two semesters.

Both experimental groups were asked to complete 37 assignments over the course of 12 weeks. 10 days were meant for chapter quizzes and exams. Other details like student population comparison of the groups, determination of germane load mechanism can be found in (Dawar, 2021).

All groups were administered the same module quizzes and final exam, and their average scores were compared to measure the impact of this technique on overall grades if any.

Module	C1 (20 students)	E1 (22 students)	E2 (20 students)
1	71% (3.72)	75% (2.05)	75% (2.22)
2	79% (2.08)	71% (2.33)	78% (3.32)
3	73% (3.19)	73% (2.55)	73% (3.68)
4	62% (3.72)	66% (2.49)	71% (3.01)
5	74% (4.26)	75% (2.44)	75% (3.10)
6	67% (3.41)	67% (1.78)	76% (1.95)
7	56% (3.48)	65% (2.50)	61% (3.30)
Average	68% (3.40)	70% (2.30)	73% (2.94)

Table 1: Mean grade points (with standard deviations) scored on the quiz by all groups

As shown in Table 1, seven chapters/modules were taught to all the groups. A quiz was given at the end of every chapter. Columns C1, C2, and E2 depict the average class scores (with standard deviations) of the quiz. The final exam consisted of a quiz that covered all seven modules, and a Java problem. Table 2 shows the average achieved by the class in the final exam.

Though there was no significant difference between module quiz scores (see Table 1), the experimental groups performed much better in the final exam (Table 2).

Even though the gains in the final quiz are marginal, the experimental groups outperformed the control group by 20 percentage points or more in JAVA program writing. The overall cumulative improvement in the final exam mean score was 16% and 19% for E1 and E2, respectively.

These numbers may insinuate that – for the experimental groups – the increased practice led to an improvement in final exam score, though it is too early to say anything with a high degree of confidence due to such a small sample size. Nevertheless, the final exam numbers are encouraging.

Group	Average Final Quiz Score	Average JAVA Program Score	Cumulative Average
C1	66%	51%	56%
E1	74%	71%	72%
E2	78%	74%	75%

Table 2: Final exam score for all groups

An end-of-course survey (see Appendix C) was conducted for both E1 and E2. The number of participants was 22 and 13 respectively, i.e., 35 students in total. One of the questions asked the students about how they felt about the utility and effectiveness of this intervention in completing the course satisfactorily. A surprising 90% of the students in E1 and 84% in E2 answered that they felt positive/better about using this technique, while 10% in E1 and 9% in E2 reported that they felt slightly worse while working with this technique.

A cumulative 45% of the students answered that working every day on assignments made it easy for them to manage stress. Students remarked that the process made it easy to manage overall stress as the assignments were gradually increasing in difficulty. 39% said it increased their stress levels as they had to do many more assignments, and 15% choose that it made no difference. The final exam results, along with the student survey responses, instilled confidence in the instructor that this technique was worth exploring.

There was one glaring and unavoidable cost of these improvements – instructor overload.

A Case for the Conversational Agent

The improvements in final exam scores, though encouraging, came at a high price as far as the instructor load was concerned. The frequency of questions asked increased in number, indicating more students were interested in asking questions. Replying to these questions consumed a significant amount of time. This load grew as the course progressed because assignments were due almost every day of the week and had to be graded quickly to provide timely feedback to students. Since every assignment was built on top of the previous one, delayed grading could mean students had no previous feedback available while attempting the current assignment. This delay is just not an option when working with AAAD. Hence, it can be seen how quickly the instructor load can increase to the point of exhaustion.

There was undoubtedly a need for support structures for the instructor. One way would be to hire a dedicated tutor and a grader. However, many instructors, due to numerous reasons, do not have access to such support. Another way would be to create a scripted expert system containing scripted question-answers. The script is a decision tree modeled by domain experts that determines which path to take in response to a question. These are static systems that may be unsuitable in circumstances where a single question can be asked in multiple ways.

Instead, a Natural Language Processing (NLP) based conversational agent/chatbot capable of answering course-related questions is chosen for bot construction in this work. The reasons for implementing such a conversational agent are multifold:

1. Many students ask the same question in different ways: Questions asked by students may be divided into two parts; text-based and knowledge-based (Scardamalia & Bereiter, 1992). Text based questions refer to queries generated as part of reading a text, while knowledge-based questions are generated through a deep interest in the topic to extend knowledge. Through the years of teaching introductory programming courses observed, the author of this work observed that many questions asked by multiple students were text-based and strikingly similar. In those cases, only the semantics and structure of the question differed, while the context of the question was the same. Hence, a system capable of understanding the context of a text based question could effectively classify multiple questions from multiple students into the same bucket and respond with a specific predefined answer. Directing these questions to an NLP-based conversational agent can save the instructor much time, which can be utilized in other areas such as mentoring. Predefined responses may not be suitable for knowledge-based questions, though.

2. Quick resolution of trivial queries: Many text-based questions asked by the

students are simple and straightforward in nature. These can be easily handled by the conversational agent, saving precious time.

3. Student's expectation of a quick response: Interaction between instructor and student is critical for student success, more so in an online environment (Chang 2009). Many studies (Li et al., 2010; Chang et al., 2015) have confirmed that students prefer asynchronous modes of communication like email or chat while interacting with instructors. A well-designed conversational agent can easily fulfill this task. Given these findings and the author's own experiences in the classroom, it is opined that the quicker a query is resolved, the stronger the student's conviction there is merit in asking questions, as they will be resolved quickly. This could lead to a reinforcement loop, making students more comfortable asking questions.

4. Long-term potential: As society goes increasingly digital, the current model of fixed classrooms, printed textbooks, and static lectures clearly fall short of fulfilling the expectations the society has of the educational establishment. Digital generation tends to learn at short or twitched speeds through parallel processing while simultaneously connected to others (Beavis, 2010). It is reported that students learn more when they immediately apply what they learned and receive help from human tutors who respond quickly (Colvin, 2007; Anwer et al., 2015). A conversational agent which is always ready to respond to student queries can be a great add-on in the toolkit of instructors.

Given all these factors, it was decided to pursue the integration of a conversational agent with the AAAD technique to create CASSA.

3. SYSTEM DESIGN

Figure 2 presents an abstracted view of CASSA. The student initiates a query through a text dialogue/message. If the conversational agent is capable of answering the query, it is annotated as "Simple," and the response is returned. Otherwise, the query is automatically sent to the instructor via email through the agent and is annotated as "Complex." When the instructor is notified of an unanswered query, they update the knowledge base of the conversational agent with a potential response while relaying the same answer/solution to the student.

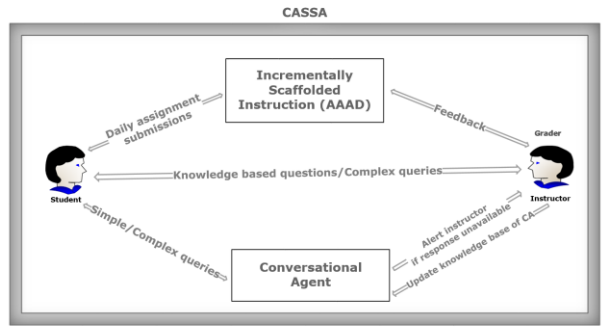


Figure 2: CASSA – An Abstraction (see Appendix B for expanded view)

Design Considerations

The retrieval process of many modern conversational agents makes use of advances in machine learning in which responses are based on predefined rules as well as analysis of the web searches. Some prominent contemporary examples are Amazon's Echo, Microsoft's Cortana, and Apple's Siri to name a few (Weinberger, 2017). The agents on the other side of the spectrum use generative algorithms and assemble responses using statistical machine translation techniques. One popular example of such mechanisms is Seq2Seq, which uses recurrent neural networks (RNN's) to accomplish the response generation.

For this work, the former approach of predefined rules aided with natural language processing algorithms was chosen. There are at least three reasons for this choice:

a) The landscape of questions asked by students in a particular course may be large, but the questions would certainly be limited by the domain of the course. This can be achieved through rule-based or information retrieval methods more efficiently since generative methods tend to be reasonably much more complex to construct.
b) By defining a rule-based template, it would be a lot easier to use the same template as a basis for another course, thereby possibly achieving re-usability in the future.
c) Generative algorithms like Seq2Seq and systems that use them tend to be relatively complex in construction and operation. Hence, it was deemed fair to use a rule-based system as a pilot.

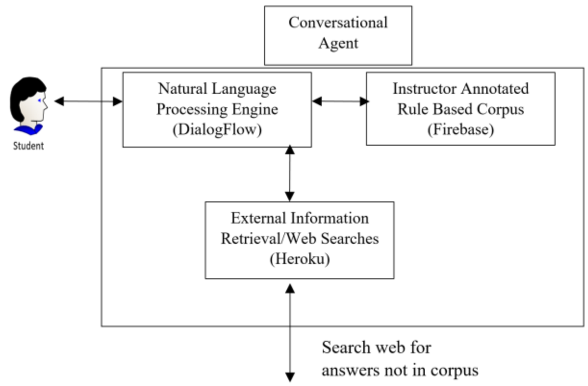


Figure 3: Conversational Agent Architecture

Figure 3 presents an abstracted view of the conversational agent used in this work. Its sub-parts are discussed below.

a) Student: Students can initiate a dialogue through three interfaces – Instructor provided web link, Dialogflow messenger, and Telegram. The student's questions are presented to the natural language processing (NLP) engine of the conversational agent (CA). It is assumed that in this day and age, students have access to the internet and should have the ability to initiate a conversation from an interface of their choice. More integrations like Facebook Messenger, Slack are possible in the future.
b) Natural Language Processing Engine (NLP): NLP can be defined as manipulation of natural language like text or speech, using mathematical representations and software. The main goal of any NLP system is to take in an unstructured input and provide a structured output. This work makes use of Dialogflow, a Google product, and a commercially available NLP platform for developing chatbots. It provides a powerful natural language processer capable of handling contextual conversations. It uses deep parsing techniques and is mainly used as integration between a conversational interface (Telegram, Slack, etc.) and the chatbot.
c) Knowledge Base: The accuracy and final employability of conversational agents depend greatly on the quality and quantity of training data. This statement is true for both generative (machine learning classifiers) and information retrieval (or rule-based) agents.

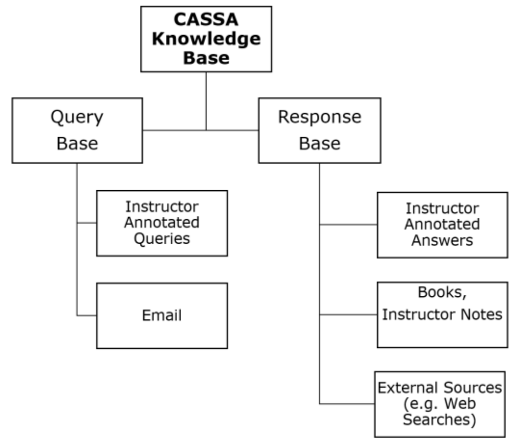


Figure 4: Conversational agent knowledge base

Though there are many ways of collecting, storing, and using the training data, this work relies upon a simplified version depicted in Figure 4.

I. Query Base: The instructor – to some extent - predefines what questions students are likely to ask in the course and creates a data set of such question-answer pairs. All the possible questions that might lead to the same response are coded under an Intent, and every Intent will have multiple questions/user examples under it. Basically, an intent categorizes the user’s intention, and the agent contains possible hundreds of such intents (231 in this work). When a user writes or says something, the NLP engine (Dialogflow in this case), matches the user expression with the best Intent. Students are also very likely to ask questions over email. This can act as a rich source of query data that the agent would need to improve its accuracy of response. This work also verifies the fact that on a single topic, many students ask the same question in different ways and formats.
All these similar questions can be represented by the same Intent to generate a single, unified response. Figure 5 illustrates this process.

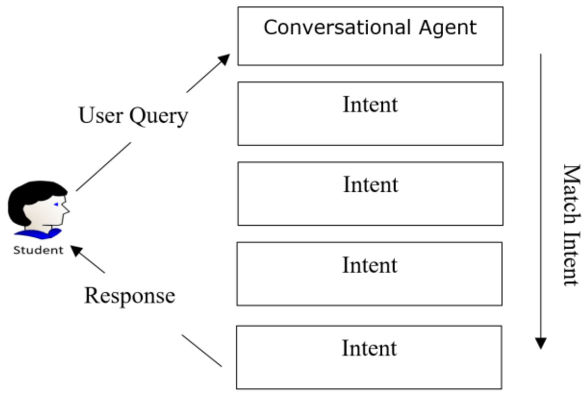


Figure 5: Intent matching

II. Response Base: This is where the responses/answers to the queries are stored. The responses to simple to mildly complex queries are stored in instructor annotated form, i.e., Intents where the instructor predefines the answer for a set of queries. The secondary source is searched if the response is not found in Intents, which includes textbook and instructor notes. Failing to find an answer in the first two sources, the agent sends the query to an external webhook.

d) External Search/Data Retrieval: The agent, as a last resort, also has the ability to query the web if it determines that a suitable response may not be present within it. This service was hosted on a web hosting platform named Heroku. Currently, only Google searches are supported. The agent extracts the relevant entities from the student's query, forms a search string, and relays it over to Heroku - a container based Platform as a Service (PaaS) - which runs a node.js service with Google search API enabled. Google search API responds with multiple URLs, and the first three URLs are presented to the student as a reference. This is not a sophisticated functionality at all. Students could easily search the web themselves and see the same URLs listed. The intention is to minimize student distraction; keep students engaged with the agent, and improve the agent's knowledge base. This query is moved to instructor annotated answers later on.

4. Conversational Agent Preliminary Evaluation

Evaluation of a chatbot is a complex problem. Many perspectives and methods, many of them subjective and often conflicting, can be utilized for its evaluation. For example, a chatbot can be evaluated on the basis of:

1. User experience
2. Information Retrieval Performance
3. Linguistic accuracy
4. Business perspective

As a direct result of a multitude of evaluation methods, numerous metrics, not necessarily mutually inclusive, have been proposed. SASSI, PARADICE, MIMIC are but a few such evaluation systems (Venkatesh et al., 2018). Some are lenient in awarding scores, while others are punitive. For example, Walker et al., 1997, proposed an attribute value matrix (AVM) to measure chatbot effectiveness. In this method, a script is created and is run through the chatbot. The desired responses are cataloged in a "scenario key," while the bot responses are recorded in the AVM. A confusion matrix (M) is then constructed as:

$$\kappa = \frac{P(A)-P(E)}{1-P(E)} \quad (1)$$

where:
P(A) = proportion of AVM aggress with the correct response
P(E) = probability of agreement by chance
κ = kappa coefficient; bot that provides random answers, κ=0; for a human κ would ideally be 1.

Other subjective methods of chatbot evaluation are presented in other studies on chatbots (Bates, & Ayuso, 1991), (Kuligowska, 2015). It becomes readily evident that no single system is able to deliver a universal framework for chatbot evaluation. Moreover, catering to so many different perspectives is an expensive endeavor and out of the scope of this work. Hence, this work focuses on the evaluation of the chatbot from the perspective of information retrieval performance only.

Specifically, this work uses a confusion matrix similar to the one suggested by Walker et al., 1997, but instead of using κ as a metric, precision, recall, and F1-scores are calculated to evaluate the chatbot.
A confusion matrix visually answers questions like - when a student asks a question X which has an actual answer Y, what was actually predicted?

The expected Intents are shown as rows, and the predicted Intents are shown as columns.

Excepted Intent/Predicted Intent	IDE_type_and_version	Filename_submission_standard	Download_IDE	Commenting_programs	Getting_started_with_java	What_is_pseudocode	What_is_flowchart	Grading_scheme	Assignment_1_1_walkthrough	Assignment_1_2_walkthrough	Assignment_1_3_walkthrough	Quiz1_q1_explanation	Quiz1_q2_explanation	Quiz1_q3_explanation	Quiz1_q4_explanation	Quiz1_q5_explanation	Precision	Recall	F1-Score
IDE_type_and_version	7		1														0.8750	0.8750	0.8750
Filename_submission_standard		7				2											1.0000	0.7778	0.8750
Download_IDE	1		5		1												0.8333	0.7143	0.7692
Commenting_programs				5	1												1.0000	0.8333	0.9091
Getting_started_with_java					6												0.6667	1.0000	0.8000
What_is_pseudocode						6									1		0.7500	0.8571	0.8000
What_is_flowchart							6								1		0.7500	0.8571	0.8000
Grading_scheme								6									1.0000	1.0000	1.0000
Assignment_1_1_walkthrough							2		11	2	1						0.7857	0.6875	0.7333
Assignment_1_2_walkthrough					1				2	10	2						0.7692	0.6667	0.7143
Assignment_1_3_walkthrough									1	1	11						0.7857	0.8462	0.8148
Quiz1_q1_explanation												10	2	1			0.9091	0.7692	0.8333
Quiz1_q2_explanation												1	11		1		0.7857	0.8462	0.8148
Quiz1_q3_explanation													1	9			0.8182	0.9000	0.8571
Quiz1_q4_explanation															10	2	0.8333	0.8333	0.8333
Quiz1_q5_explanation														1	1	11	0.8462	0.8400	0.8431

Figure 6: Confusion Matrix for Module 1

Figure 6 shows the confusion matrix for module 1 that has 16 Intents. Every Intent has multiple user examples, which are nothing but different ways of asking the same question. For example, a student can inquire about the IDE for the course. There are many ways this question be asked. Some of them are:

1. What is the IDE we are using?
2. What's the IDE name?
3. Can I use Netbeans IDE?
4. Tell me the IDE for this course?
5. What is the software to run Java programs?
6. What is the software we are using for this course?

These user examples are sent to the agent, and whenever the expected and predicted Intent is a match, the diagonal cell value is increased by 1, and these are called successful test cases. All other cell values that are not on the diagonal are failed test cases. Again, it must be noted that this method leaves out many other vital facets like evaluating chatbot looks, appearance, personality. These aspects may be evaluated in the future as the work on this system progresses.

At the time of writing, the agent had access to had 231 instructor annotated Intents, instructor class notes compiled as a .pdf, and a freely available Java textbook as a .pdf. Out of the 231 Intents, 104 were predefined by the instructor, and the rest were compiled from the questions asked by students on email over years of teaching this programming course. It should be noted that every Intent contains examples/queries that are written in different formats/ways but point towards the same response/answer. The distribution of Intents among different chapters/modules is listed in Table 3.

Module	No. of Intents
1	16
2	27
3	37
4	40
5	41
6	39
7	31
Total	231

Table 3: No. of Intents per module

The instructor annotated Intents if correctly matched with the user query, are the first line of response. If the response isn't found in those intents, the query is referred to instructor notes or the textbook, and then the web, in that order. The more such intents the agent has access to, the better the potential accuracy of the agent. Ideally, the number of intents should progressively expand as the course is taught multiple times over, and the new questions by the students, and previously unknown questions to the agent, are fed into the knowledge base.

Three performance metrics, namely precision, recall, and F1-score, were measured for every Intent. As can be seen, there are numerous ways of asking the same question. These ways are the instructor annotated queries or user examples. All these questions should match the same Intent, which in this case should be IDE_type_and_version. However, it is tough to achieve such perfect performance. For the sake of brevity, Figure 6 only displays the performance of the agent for Module 1 having 16 intents. The precision, recall, and F1-score are also shown in the three rightmost columns.

Averages of all 231 Intent performance scores were computed to mark the final performance measures of the agent. The results are listed in Table 4.

Performance Metric	Average Metric Scores for Seven Modules
Precision	0.7981
Recall	0.7856
F1-Score	0.7923

Table 4: Preliminary performance score of conversational agent

F1-Score below 0.80 is less than desirable, and F1-Score above 0.90 is considered good.

As a work in progress, the author believes that an F1-Score of 0.7923, though only slightly comforting, is a reasonable milestone in the preliminary agent development while acknowledging that a lot more training data and improvements are required to make this agent usable in live courses. See Appendix B for example conversations between the chatbot and a student. The integration with Dialogflow Messenger and Telegram is shown.

5. CONCLUSION AND FUTURE WORK

At an anecdotal level, the results indicate that it may be possible to affect the motivation levels of novice programmers using incrementally scaffolded instruction. Though there were no significant differences in the individual chapter quiz scores between the control and experimental groups, the experimental groups performed significantly better in the final exam. This came at the price of significant instructor overload. The integration of a helper chatbot with this technique is expected to reduce the instructor load. The initial preproduction performance of the conversational agent is undoubtedly below expectations but is expected to improve with more data and time. One of the ways the author intends to collect more data/user examples is to use the course chat forums and discussion boards for more questions asked by students to each other. The next step will be continuous training of the chatbot to achieve an F1-Score of at least 0.85, after which it will be opened for students to use.

To further mitigate the load on the instructor while maintaining the integrity of the technique, integrating an automatic grading system with the CASSA is proposed. An abstract schema of this system is shown in Figure 7.

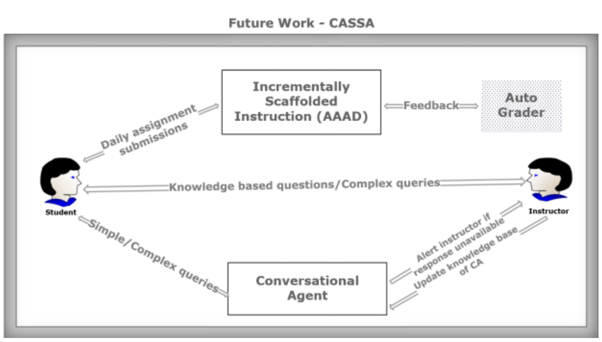


Figure 7: Integration of Auto Grader with CASSA

In closing, it would be too premature to consider the CASSA system as a workable method for affecting student motivation, given the significant challenges this system entails presently. The preliminary results, nevertheless, are encouraging and provide a solid direction for future research.

6. REFERENCES

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

Ali, N., Anwer, M., & J., Abbas. (2015). Impact of Peer Tutoring on Learning of Students. Journal for Studies in Management and Planning, 1(2), 61-66.

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

Bates, M., & Ayuso, D. (1991). A proposal for incremental dialogue evaluation. Proceedings of the workshop on Speech and Natural Language - HLT '91.

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

Beavis, C. (2010). Literacy, Learning, and Online Games:Challenge and Possibility in the Digital Age. In Proceedings of the IEEE 3rd International Conference on Digital Game and Intelligent Toy Enhanced Learning. Piscataway, NJ: Institute for Electrical and Electronics Engineers.

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

Beutel, M. E., Klein, E. M., Aufenanger, S., Brähler, E., Dreier, M., Müller, K. W., Quiring, O., Reinecke, L., Schmutzer, G., Stark, B., & Wölfling, K. (2016). Procrastination, Distress and Life Satisfaction across the Age Range - A German Representative Community Study. *PloS one*, *11*(2), e0148054.

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

Chang, C-W. (2009). Efficacy of interaction among college students in a web-based environment. Journal of Educational

Technology Development and Exchange, 2(1), 17-32.

Colvin, J. W. (2007). Peer tutoring and social dynamics in higher education. Mentoring and Tutoring, 15(2), 15-181.

Chang, C-W., Hurst, B., & McLean, A. (2015). You've got mail: Student preferences of instructor communication in online courses in an age of advancing technologies. Journal of Educational Technology Development and Exchange, 8(1), 39-47.

Dawar, D., (2021). Towards Improving Student Expectations in Introductory Programming Course with Incrementally Scaffolded Approach. Information Systems Education Journal 19(4), 61-76.

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

Goold, A., and Rimmer, R. (2000). Factors affecting performance in first-year computing. SIGCSE Bulletin 32, 39–43.

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

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

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

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

Keller, J. M. (1983). Motivational design of instruction. In Instructional-Design Theories and Models: An Overview of their Current Status, C. M. Reigeluth, Ed. Lawrence Erlbaum Associates, pp. 383–434.

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

Kim, K. R. & Seo, E. H. (2015). The relationship between procrastination and academic performance: A meta analysis. Personality and Individual differences, 82, 26-33.

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

Kuligowska, K. (2015). Commercial Chatbot: Performance Evaluation, Usability Metrics and Quality Standards of Embodied Conversational Agents. Professionals Center for Business Research, 2(02), 1-16. doi:10.18483/pcbr.22

Li, L., Finley, J., Pitts, J., & Guo, R. (2010). Which is a better choice for student faculty interaction: Synchronous or asynchronous communication? Journal of Volume 9, No. 1, September, 2016 11 Technology Research, 2, 1-12.

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

Moors, A., & Houwer, J. D. (2006). Automaticity: A Theoretical and Conceptual Analysis. Psychol Bull, 132(2), 297-326.

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

Paas, F., Renkl, A., & Sweller, J. (2010). Cognitive Load Theory and Instructional Design: Recent Developments. Educational Psychologist, 38 (1), 1-4.

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

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

Scardamalia, M. and Bereiter, C. 1992. Text-based and knowledge-based questioning by children. *Cognition and Instruction*, 9: 177–199.

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

Steel, P. (2007). The nature of procrastination: A meta-analytic and theoretical review of quintessential self-regulatory failure. *Psychological Bulletin, 133*(1), 65–94.

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

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

Venkatesh, A., Khatri, C., Ram, A., Guo, F., Gabriel, R., Nagar, A., Raju, A. (2018). On Evaluating and Comparing Conversational Agents. ArXiv:1801.03625 [Cs].

Walker, M. A., Litman, D. J., Kamm, C. A., & Abella, A. (1997). Paradise. Proceedings of the 35th annual meeting on Association for Computational Linguistics.

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

Weinberger, M. (2017). Why Amazon's Echo is totally dominating - and what Google, Microsoft, and Apple have to do to catch up.

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

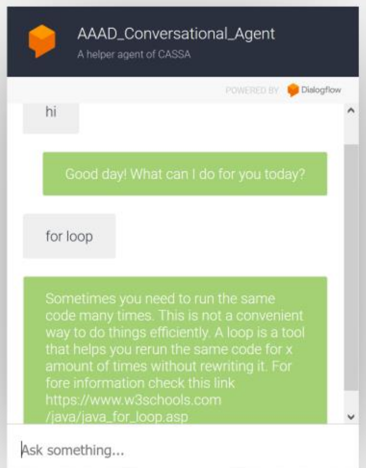
APPENDIX A

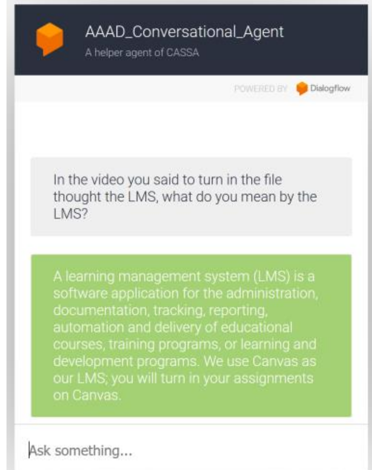
Table X: Increment in cognitive load with time

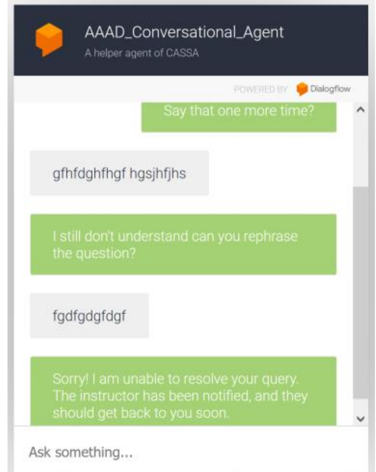
Assignment No.	Description	Concepts Tested	Cognitive Load
1	Write a method printS that takes a string as an input and prints it to the console.	Rudimentary method writing.	Low
2	Modify the above method printS and enable it to take another argument, an integer, *n*. The method then prints the string *n*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

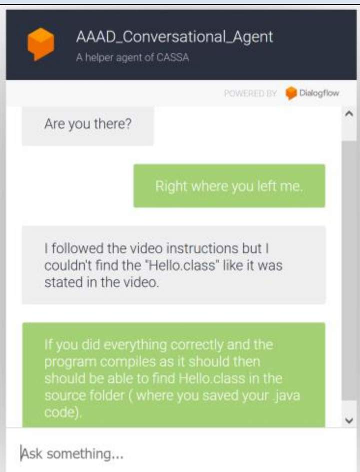
APPENDIX B

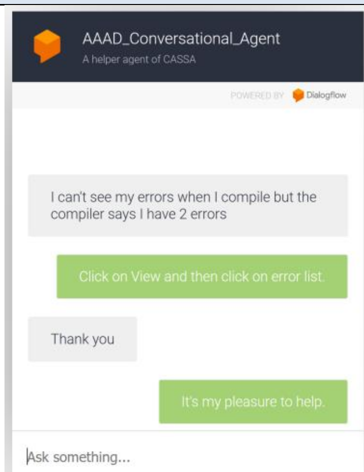
Dialogflow Messenger Integration

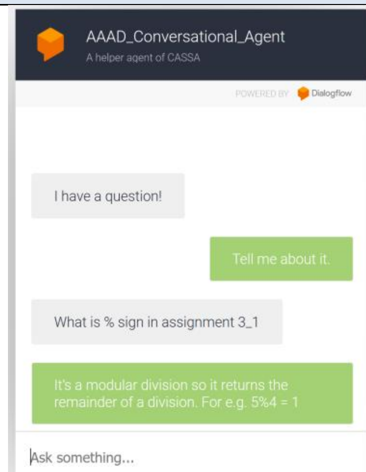


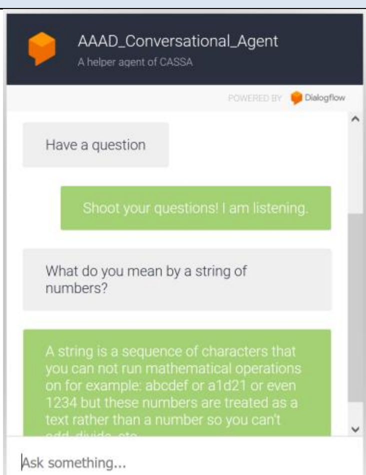


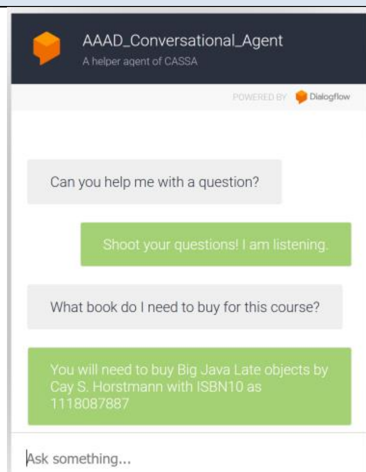


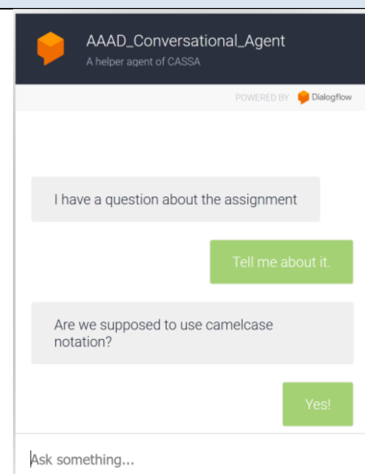




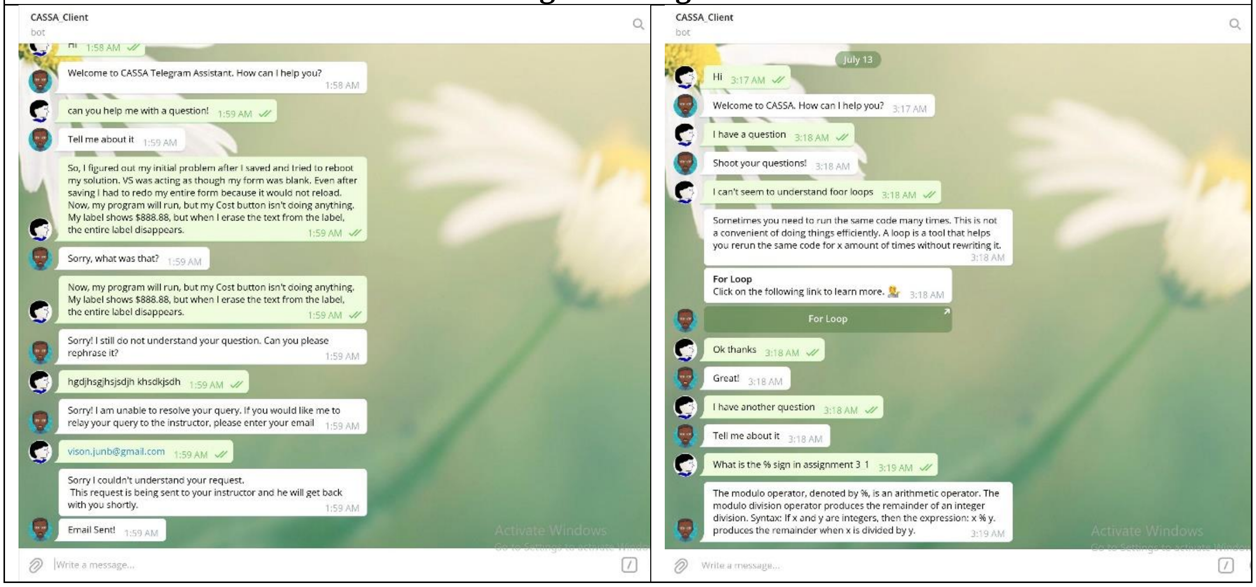




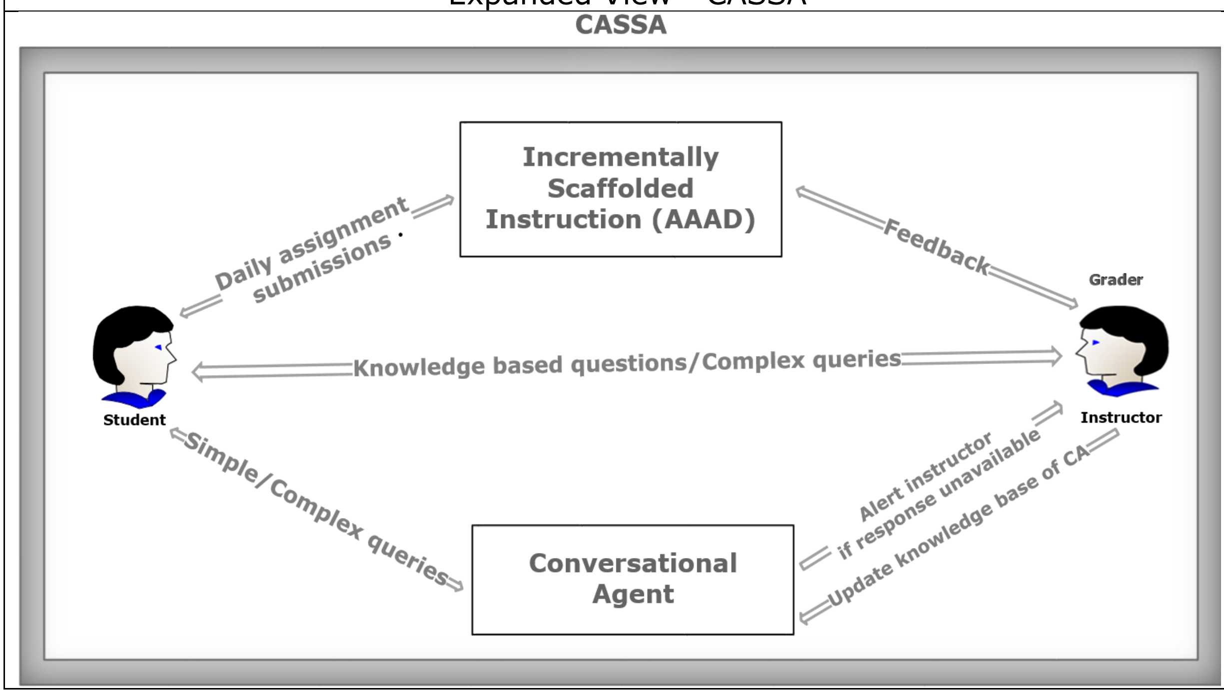




Telegram Integration



Expanded View - CASSA



APPENDIX C

CSE 174: Student experiences with multiple assignments

English ▼

SURVEY INSTRUCTIONS

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

Student Resources

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

Definitely yes	Probably yes	May be	Probably not	Definitely not

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

Definitely yes	Probably yes	May be	Probably not	Definitely not

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

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

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

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

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

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

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

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

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

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

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

One long and possibly difficult assignment each week

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

No preference

Block 2

How did doing multiple assignments effect your stress levels?

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

It increased my stress as I had to do many assignments

It made no difference

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

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

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

A Topical Examination of the Introduction to Information Systems Course

Kevin Slonka
slonka@pitt.edu
Information Sciences Department
University of Pittsburgh
Greensburg, PA 15601 USA

Neelima Bhatnagar
bhatnagr@pitt.edu
Information Sciences Department
University of Pittsburgh
Greensburg, PA 15601 USA

Abstract

The introductory Information Systems (IS) course is a critical course at the beginning of every IS/IT major's degree. While many textbooks exist that are focused solely on this course, they all vary in the covered topics and the depth of material. This research examined most major textbooks focused on the introductory course and, through qualitative and quantitative analysis, reached a final set of 14 themes that should be taught in this course. Implications are drawn and the groundwork for future studies are laid.

Keywords: Information Systems, introductory, foundations, textbook, competency, knowledge

1. INTRODUCTION

The introductory course in information systems continues to be a foundational course in the discipline. While there is an abundance of textbooks published on the subject matter used to teach the course, the core concepts covered in each text varies greatly. Despite the fact that "information systems touch almost every aspect of students' lives, […] students are often detached and uninterested in the introductory course" (Golub, 2015, p. 442). The dilemma of how to engage students, both majors and non-majors, in the subject-matter is an important problem that has not been examined in prior research.

The intro course is a major requirement for Information Technology (IT) majors as well as Management Information Systems (MIS) majors at a regional campus of an R1 university in Western Pennsylvania. In addition to those seeking IT or MIS degrees, students from non-technology majors also enroll in this course. This initial study is focused on eliciting from literature the core learning activities that should be present in an introductory course, leading to the sole research question:

RQ 1: What are the core knowledge areas that should be present in an introductory IS/IT course?

2. THE INTRO COURSE CONCEPT

Virtually every major has some form of introductory course: from slow-paced courses that ease students into a subject matter to hard-hitting "gatekeeper" courses (Gasiewski, Eagan, Garcia, et al., 2012; Mervis, 2010) that are created to ensure only the best students progress to the next semester. Some courses are lucky enough to have textbooks published specifically for an introductory level of content while other courses get by without a textbook or any standard pedagogy (Cohen & Wang, 2016). One thing is clear, there are many challenges to teaching introductory courses: some "related to the students' characteristics, the teaching methods, or the nature of [the content]" (Alammary, 2019, p. 2).

Schwartz & Smith (2010) detail one of the largest issues for introductory courses: large lectures. When enrollment in courses exceeds 200 students, that, along with other intimidating factors, such as large lecture halls, make any kind of interaction between students or professors difficult. This only gets worse when universities do not allow room in the budget for recitation/discussion sessions, ensuring that "the course almost inevitably devolves into the dull lectures that fuel universal discontent" (p. 250). Their approach was to lessen the breadth of the content so that the more important concepts in the field can be examined. Additionally, certain pedagogical aspects were introduced that had success, even in courses with over 200 students. In-class surveys (typically by a show of hands), assignments that applied directly to life so that students could see "concrete manifestations" (p. 263), online discussion threads, and short pop quizzes were four tools that had the most success.

Many studies have been conducted focusing on "gatekeeper" courses and ways to present them such that more students complete the courses, and furthermore, their degrees. One such study found that only 33% of white students and 42% of Asian-American students actually complete their degree in 5 years. After overhauling their introductory courses, one university had a 15% increase in pass rates (Mervis, 2010). Such changes as replacing recitation sessions with small group problem solving sessions, adding teaching assistants, mandating attendance, and ensuring that remaining lecture sessions are taught by professors instead of grad students are just some of the changes that ushered in the increased pass rates.

Other studies have focused on the learning models used in teaching introductory courses. Such models as the flipped model, mixed model, flex model, supplemental model, and online-practicing model have undergone various levels of scrutiny. According to Alammary (2019), the mixed model, where mini-lectures and course assignments happen both online and in-person, contributes to better student performance.

Allos, Yakes, Fleming, Cutrer, Pilla, Clair, Fowler, & Miller (2018) added that "lasting impact[s] on students' attitudes and beliefs" (p. 1310) could be made by adding humanistic concepts to the course; "thoughtful and deliberate introductions to the profession" (p. 1310). Instead of merely teaching theory in the classroom, students should be introduced to the actual places in which their profession occurs. For medical students, this would mean trips to clinics. For IT students, this could mean trips to data centers.

3. THE INTRO IS/IT COURSE

Modaresnezhad & Schell (2019) argue that "students of all [technology-related] majors need to be given the foundation knowledge and then carry the appreciation for information systems into their careers" (p. 39-40). The introductory course, "in part because of its wide breadth and lack of depth, […] remains one of the more difficult and challenging courses for faculty to teach" (Holmes, 2003). The textbook one chooses to use for such a course is critical due to its role for introducing the student to and focusing the core IS/IT topics (Hassan & Becker, 2007; 2003).

A comparison of the IS 2010 and 2020 model curricula revealed that Foundations of Information Systems continued to be listed as a core course. This course:

> refers to the ability of students to understand the fundamental concepts of IS (including hardware, software, and information acquisition) and the support that IS provides for transactional, decisional, and collaborative business processes. They will also be able to understand the collection, processing, storage, distribution, and value of information and be able to make recommendations regarding IS that support and enable individuals in their daily lives as well as the management, customers, and suppliers of the enterprise. This competency includes the ability to conduct and organizational business analysis, and assess processes, and systems (ACM, 2020, p. 51).

Also of note is the method of instruction or other properties of the intro IS course. While much education literature touts the benefits of cooperative education, some studies have found that there was no positive effect on learning outcomes with the method (Wehrs, 2002). Other studies found that certain activities such as assigning non-IS authors to tell the IS story as required reading, utilizing a plethora of writing assignments, and exposing the students to various exciting technologies had positive effects on their classes (Whelan & Firth, 2012). What is common amongst most scholars is the belief that the introductory course should have a wide breadth and that this course specifically impacts how students view the IS field (Akbulut, 2015).

4. METHODOLOGY & RESULTS

This research utilized a combination of qualitative and quantitative methods to elicit the required knowledge areas of the introductory Information Systems course.

An initial survey of the educational material landscape revealed nine prominent textbooks from highly regarded publishers aimed at the introductory IS course. Table 1 lists the texts that were analyzed in this process.

Title	Publisher
Information Systems	Chapman & Hall
Information Systems	FlatWorld
Information Systems for Business	Prospect Press
Introduction to Information Systems	MyEducator
Introduction to Information Systems	Pearson
Introduction to Information Systems	Wiley
M: Information Systems	McGraw Hill
Managing and Using Information Systems	Wiley
Principles of Information Systems	Cengage

Table 1: List of Analyzed Textbooks

A list of major topics from each textbook was recorded and inductively coded into major themes. The final themes list was arrived at through three rounds of coding. 32 themes emerged from the data. In order to narrow the focus to only those deemed critical, a frequency analysis was conducted on the themes. There was a frequency cutoff for inclusion in the final thematic list. This was done to ensure that only as many topics as can reasonably be taught in a standard, 15-week semester were selected. With this in mind, the frequency cutoff for the number of textbooks in which a topic was covered was five. This limited the list to 11 themes, which accounted for 66% of the total, as shown in Table 2.

Further analysis was conducted on the thematic data by comparing it to the core knowledge areas from three ACM model curricula: Information Systems 2010, Information Systems 2020 (Dec. draft), and Information Technology 2017. Understanding that solely relying on textbook publishers to tell faculty what they should be teaching in their courses is not prudent, these three model curricula most closely align to the majors in which an Intro IS course would be found.

Such a course would not be found in three of the five computing curricula maintained by ACM: computer engineering, computer science, and software engineering. Therefore, only the IS and IT model curricula were used as they lay the foundation to educate computing professionals who can "select, develop, apply, integrate, and administer secure computing technologies to enable users to accomplish their [...] goals" (ACM, 2017, p. 18) or support a business's "transactional, decisional, and collaborative business processes" (ACM, 2020, p. 52) from a hardware, software, and information acquisition standpoint.

These comparisons of the themes to the core knowledge areas of model curricula revealed the areas of each curriculum that were covered by the critical themes, thus lending credence to the theme as a topic critical to the Intro IS course.

Theme	Freq.	Cum. %
Security	12	9%
Analytics	11	17%
Data/DB	11	25%
Development	11	34%
Organization	7	39%
Overview	7	44%
Strategy	7	49%
Commerce	6	54%
Networking	6	58%
Acquisition	5	62%
Social	5	66%

Table 2: Critical Themes

As shown in Table 3, the 11 critical themes cover only five of the domains from the ACM Information Technology 2017 model curriculum. This, however, is more than the single knowledge area of coverage from the ACM Information Systems 2010 model curriculum, listed in Table 4.

Domain	Coverage
Information Management	
Integrated Systems Technology	
Platform Technologies	
System Paradigms	
User Experience Design	
Cybersecurity Principles / Cybersecurity Emerging Challenges	X
Global Professional Practice / Social Responsibility	X
Networking / Applied Networks	X
Software Fundamentals / Software Development and Management	X
Web and Mobile Systems / Mobile Applications	
Cloud Computing	
Data Scalability and Analytics	X
Internet of Things	
Virtual Systems and Services	

Table 3: IT 2017 Model Curriculum Coverage

As of December 2020, the IS model curriculum is undergoing revision and is in draft form. This update moves away from the previous Knowledge Areas to a set of required Competency Areas, which more closely align to the prescribed list of courses from the 2010 model. Listed in Table 5, the critical themes cover six of the 10 new Competency Areas (though one Area is a practicum that typically is not covered by any textbook).

Knowledge Area	Coverage
IS Management and Leadership	
Data and Information Management	X
Systems Analysis & Design	
IS Project Management	
Enterprise Architecture	
User Experience	
Professional Issues in Information Systems	

Table 4: IS 2010 Model Curriculum Coverage

In summary, the critical themes cover 36% of the IT 2017 domains, 14% of the IS 2010 areas, and 60% of the IS 2020 areas.

5. DISCUSSION

This research produced many interesting findings. The comparison of the critical themes to the three model curricula was surprising. While the focus of this research was Information Systems, many university programs offer a mix of IS and IT. The critical themes covered more topics from the

Competency Area	Coverage
Foundations of Information Systems	X
Data/Information Management	X
IT Infrastructure	X
Secure Computing	X
Systems Analysis & Design	
Application Development / Programming	X
Ethics, Use, and Implications for Society	
IS Management & Strategy	X
IS Project Management	
IS Practicum	

Table 5: IS 2020 Draft Model Curriculum Coverage

2017 Information Technology curriculum than the 2010 Information Systems curriculum. This leads one to believe that the introductory IS course is not strictly in the IS silo but consists of a broad arrangement of topics in order to be used as an entrance course for different technology-focused programs.

When analyzing the 2010 IS curriculum along with the draft 2020 curriculum, a shift in terminology and content arrangement is noticed. The concept of Knowledge Areas from the 2010 curriculum is no longer present in the 2020 curriculum. The newer model takes what used to be simply a list of courses and massaged them into broader Competency Areas. These Areas are more similar to the Domains in the IT model than the Knowledge Areas in the original 2010 IS model. For this reason, a more accurate comparison is between the 2017 IT model and the 2020 IS model. It is when reviewing theme coverage between only these two models that one can see a common assumption manifest: intro IS textbooks cover topics in line with the IS model curriculum.

Focusing on the textbooks themselves, while one may have expected all textbooks created for the introductory IS course to cover the same, or at least mostly similar, topics, that was not the case. Nine textbooks produced 32 different themes. Almost half (15) of those themes only appeared in one or two textbooks. Although assumptions cannot be made about each author's intentions, it is clear that the intro course has different purposes for different authors. Some authors write their text as a general overview of the field, some wrote their text from a management perspective, and others wrote their text to focus on only a handful of critical areas, such as development. This disparity may seem in opposition to the purpose of this research; however, it offers the unique insight that there is no single way to teach the introductory IS course. Recalling the previously explained crossover between IS and IT topics, educators must be open-minded. Universities should be welcome to package the introductory course in a way that best fits their program rather than constraining the topics to one particular silo.

Theme	
Security	
Analytics	
Data/DB	
Development	
Organization	
Overview	
Strategy	
Commerce	
Networking	
Acquisition	
Social	
Management	*New*
Mobile/Cloud	*New*
Infrastructure/Hardware	*New*

Table 6: Final theme list

With that in mind, Table 6 presents an updated list of critical themes. The original list contained 11 themes elicited from the source material but did not cover all aspects of any model curriculum. Additionally, some topics that did not make the cut are crucial for anyone entering the IS or IT field.

Included in the Management theme are not only discussions from a manager/supervisor perspective but also the foundations of systems analysis/design and project management. These concepts are critical in collaborative work environments and are typically required learning in the later years of one's degree. Another skillset all but required by employers is knowledge of networking infrastructure and hardware. With many sophomore-level and higher courses assuming an understanding of the fabric on which digital business is conducted, one would be remiss not to cover such information minimally at a basic level. The last addition to the theme list used to be considered new and emerging technology but has proven itself a mainstay of the digital world: mobile and cloud computing. With the introduction of Amazon EC2 in 2006 and the modern smartphone (iPhone) in 2007, mobile and cloud computing have been prevalent for over 15 years. They can no longer be classified as new or emerging. Only being covered by two of the textbooks analyzed in this research is a disservice to the IS/IT student who needs to have a solid understanding of how these technologies fit into and shape our world.

6. LIMITATIONS & FUTURE RESEARCH

One limitation of this research is the selection of textbooks. While every effort was given to locate all current textbooks on the subject through publishers, retail outlets, literature reviews, and web searching, it is possible that some were not found. Additionally, IS courses are sometimes found in business schools and sometimes found in engineering/science schools, as is the case with the authors. Because of this the goals of this research might differ from the reader's due to the placement of their course.

This study acts as a stepping stone to the ultimate goal of determining a proper set of learning activities for the introductory IS course that will purposefully engage and stimulate the student's mind. Now that the required topics for this course are established, future studies can explore experiential learning activities. Additionally, individual case studies for implementation of this research and longitudinal studies on the effectiveness of these findings are possible.

7. REFERENCES

ACM. (2017). *Information Technology Curricula 2017*. Retrieved on May 18, 2021 from https://www.acm.org/binaries/content/assets/education/curricula-recommendations/it2017.pdf

ACM. (2020). *IS2020: Competency model for undergraduate programs in information systems, draft*. Retrieved on from https://is2020.hosting2.acm.org/wp-content/uploads/2021/06/is2020.pdf

Akbulut, A. Y. (2015). The impact of the introductory IS course on students' perceptions of IS professionals. *Journal of Information Systems Education, 26*(4), 295-304.

Alammary, A. (2019). Blended learning models for introductory programming courses: A systematic review. *PLoS ONE, 14*(9), 1–26.

Allos, B. M., Yakes, E. A., Fleming, A., Cutrer, W. B., Pilla, M., Clair, W., Fowler, M., & Miller, B. (2018). Framing Medicine as a Moral Practice. *Academic Medicine, 93*(9), 1310–1314.

Baltzan, P. (2017). *M: Information Systems, 4th edition.* McGraw-Hill Education.

Belanger, F., Van Slyke, C., & Crossler, R. E. (2022). *Information systems: An experiential approach, Edition 4.0.* Prospect Press.

Cohen, M. M. & Wang, G. (2016). Teaching the introduction to American studies course: A dialogue. *American Quarterly, 68*(2), 347-354, 499, 506.

Gallaugher, J. (2013). *Information systems: A manager's guide to harnessing technology, Edition 2.0.* FlatWorld.

Gasiewski, J. A., Eagan, M. K., Garcia, G. A. et al. (2012). From Gatekeeping to Engagement: A Multicontextual, Mixed Method Study of Student Academic Engagement in Introductory STEM Courses. *Research in Higher Education, 53*, 229–261.

Golub, B. (2015). Teaching intro IS with a learner-centered, experiential approach. 2015 Proceedings of the Information Systems Education Conference (vol. 32), Orlando, FL.

Hassan, N.R., & Becker, J.D. (2007). Uncovering conceptual gaps in introductory IS textbooks. *Journal of Information Systems Education, 18*(2), 169-182.

Keith, M., Gaskin, J., & Dean, D. (2021). *Introduction to information systems.* MyEducator.

Mallach, E. G. (2015). *Information systems: What every business student needs to know.* Chapman & Hall.

Mervis, J. (2010). Better Intro Courses Seen as Key to Reducing Attrition of STEM Majors. *Science, 330*(6002), 306-306.

Modaresnezhad, M. & Schell, G. (2019). The soul of the introductory information systems course. *Information Systems Education Journal, 17*(5), 39-46.

Pearlson, K. E., Saunders, C. S., & Galletta, D. F. (2015). *Managing and using information systems: A strategic approach, 6th edition.* Wiley.

Rainer, R. K. & Prince, B. (2017). *Introduction to information systems, 7th edition.* Wiley.

Schwartz, M., Smith, R. T. (2010). Beyond the Core: The Hot Topic(al) Alternative to the Survey-Based Introduction to Sociology Course. *American Journal of Sociology, 41*, 249–276.

Stair, R. & Reynolds, G. (2021). *Principles of information systems.* Cengage.

Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K. M., Nunamaker, Jr., J. F., Sipior, J. C., de Vreede, G. J. (2010). IS 2010: Curriculum guidelines for undergraduate degree programs in information systems. ACM. https://www.acm.org/binaries/content/assets/education/curricula-recommendations/is-2010-acm-final.pdf

Wallace, P. (2015). *Introduction to information systems, 2nd Edition.* Pearson.

Whelan, E. & Firth, D. (2012). Changing the introductory IS course to improve future enrollments: An Irish perspective. *Journal of Information Systems Education, 23*(4), 395-406.

Whers, W. (2002). An assessment of the effectiveness of cooperative learning in introductory information systems. *Journal of Information Systems Education, 13*(1), 37-50.

Editor's Note:

This paper was selected for inclusion in the journal as an EDSIGCON 2021 Meritorious Paper. The acceptance rate is typically 15% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2021.

A Comparison of Student Perceptions and Academic Performance across Three Instructional Modes

Vic Matta
matta@ohio.edu
Ohio University
Athens, OH

Shailendra Palvia
Shailendra.Palvia@liu.edu
Long Island University
Brooklyn, NY

Abstract

The recent pandemic compelled educational institutions all over the world to shift to online instruction. And now institutions find themselves trying to answer questions like how should we proceed when we come back to normal? Which online instructional innovations should we keep? This research attempts to answer those questions by comparing three modes of instruction: in-class, synchronous, and asynchronous for the same course during a semester taught by one instructor. The research analyzes responses from students on the following instructional characteristics: instructor involvement, interaction amongst students, interaction with instructor, course design, student satisfaction and learning experience. Survey data is analyzed using a repeated measures design with pairwise comparisons to understand the differences in students' perceptions of instructional characteristics across these modes. The study also explores differences in actual learning outcomes. Results showed that students overwhelmingly perceived all instructional characteristics to be better facilitated with in-class instruction than with either of the online modes, except for course design which showed no significant differences. It is also seen that students perceive synchronous and asynchronous instruction to have many parallels. Commentary from students suggests that online instructional practice may need a shift, not just in technology improvement, but also in pedagogical design. Students noted that with the shift to online instructional modes, they would like to see increased flexibility, willingness to personalize support, and timely responses.

Keywords: instructional modes, student satisfaction, new normal, e-learning, synchronous, asynchronous, repeated measures design

1. INTRODUCTION

The mandatory shift to online education led innovations in online instructional delivery. Video conferencing platforms increased their customers many-fold and improved their offerings in online instruction. Instructors ramped up their skills of technology use and pedagogical approaches. Remote teaching lasted longer than expected and permitted all stakeholders to get better than they had during the hasty transition. For instance, videoconferencing platforms like Zoom improved their video quality and security features. They created the ability to add status updates and streamlined their application to be easy to use (Correia, Liu, & Xu, 2020) among other improvements. Instructors began to educate themselves on techniques to improve online instruction by reading or reviewing texts on the topic (Matta, 2021). Researchers had already been discussing techniques to overcome some of the limitations in e-Learning, such as maintaining student attention in synchronous sessions (Hrastinski, 2008), or reducing isolation in asynchronous sessions (Ballenger & Garvis, 2010). Students became more accustomed to online learning and the more introverted students actually preferred asynchronous education (Hood, Jacques, Chen, & Hebert, 2021).

The innovation around remote instruction generated a spectrum of instructional models between fully online to fully in-person, depending on location: whether in-class or anywhere, timing: whether simultaneous or at student's own time, tools: classroom technology, or portable technology, and finally with varying levels of flexibility. For the sake of clarification, a few salient instructional models are briefly defined. According to Kakeshita (2021), the term Hybrid is often used generically to imply some permutation of online instruction, whether it is synchronous or asynchronous. This understanding is sustained for the purposes of the current research and can therefore include HyFlex or Blended education. In his open source book, Beatty (2014) defines a HyFlex course as one in which students have a choice for attending the course in-person or online. In contrast, a blended course uses both, online and in-person modes, not one or the other. Additionally, students do not have a choice of instructional mode. While HyFlex courses provide students with a choice, HyFlex course style can be more difficult to implement, and often needs instructor training (Raman et al., 2021). An instructional support person may also be needed to facilitate the instructor's divided attention between the online and in-class student (Pathak & Palvia, 2021). Therefore, they may be best suited for times during a pandemic, but not necessarily for the new normal after the pandemic.

Both, in-class and blended instruction require physical classrooms and some version of in-person presence. Before the Covid-19 pandemic, instructors have often used blended instruction for teaching analytics because analytical techniques can be detailed and involved. Recorded video for more involved analytical techniques may be reviewed more than once. Often, analytics courses use such recorded instruction to assign an initial preparatory assignment before the first in-class discussion (Sokout, Usagawa, & Mukhtar, 2020). After a year of various hybrid versions of instructional delivery, the question arises: what does the new normal hold for us? Can we return to the old way of doing things with blended courses? The questions arise not only because instructors have refined online synchronous and asynchronous instruction, but also because students often tend to prefer online instruction even while they are on-campus (Kelly, 2021). One way to proceed is to understand students' perceptions and actual learning outcomes with online instruction as compared with that of in-class instruction.

This research informs us about these perceptions and actual learning outcomes by comparing them across in-class, online synchronous and online asynchronous modes of instruction. The rest of the paper develops the research question, and is followed by methodology of the research, discussion of results, limitations and conclusion.

2. RESEARCH DEVELOPMENT

There has been considerable research in the last decade on online education, comparing in-class, hybrid and online modes of instruction. Studies comparing student perceptions on learning, academic performance, satisfaction, level of interaction, across these modes have reported a preference for in-class instruction (Weldy, 2018). According to one study (Fish & Snodgrass, 2015) that surveyed undergraduate and graduate business students, perceptions of online instruction improved as students took more online courses. However, on occasion, students were asked about their perceptions of modes of online instruction without having taken the class. Findings of these studies are interesting but need to be interpreted with caution as students' perceptions were not based on their experience with all three modes (Weldy, 2018).

Some studies, on the other hand, have found contradictory results, with student preferences for online modes of instruction. For instance, a few studies have found higher levels of student satisfaction and perceptions of learning and engagement in online than in-class modes, in which students were enrolled in the same course taught by the same instructor (Bowers & Kumar, 2015; Fadol, Aldamen, & Saadullah, 2018). These divergent findings make it challenging to reach a clear understanding of student perceptions and satisfaction across different modes of learning. There is, therefore, a need to further investigate perceptions and experiences across modes.

Findings with student learning outcomes were also mixed. A meta-analysis of nine studies that examined differences in student performance for college level economics courses between the years 2000 and 2012 found student performance to be stronger for in-class courses, as compared with online synchronous and asynchronous courses. Another study also reported an interesting finding, that prior academic achievement was a significant moderator (Sanford, 2017). Students with prior lower academic record performed significantly better for in-class modes of instruction than for online modes, but this was not the case for students with a higher academic record. It appears that lower performing students need the in-class instruction to motivate them and generate the required discipline. This finding has been confirmed by other studies as well (Evans, 2013; Flanagan, 2012). On the other hand, studies that also examined differences in academic performance across instructional modes had more mixed results. Callister and Love (2016) examined differences in a negotiations course, while DiRienzo and Lilly (2014) examined differences between instructional modes for concepts with varying complexity. In both cases, no difference was found in learning outcomes across different modes of instruction.

Student Perceptions and Learning Outcomes

Prior research has examined student experience between instructional modes using various approaches. Ahmed (2010) surveyed students to examine acceptance of hybrid learning using information technology (IT) infrastructure, instructor characteristics, and organizational and technical support. Information Technology infrastructure and organizational support were proven to be key determinants of the instructor characteristics as a critical success factor of hybrid e-learning acceptance. In another study, Miranda, Isaias, Costa, and Pifano (2017) leveraged an extensive literature review and focus groups with different stakeholders to identify technology type, course content, students' and instructors' attributes as critical success factors for online learning. Another research study (Sebastianelli, Swift, & Tamimi, 2015) built on prior research and surveyed 169 MBA students to find that course characteristics, interaction amongst students and interaction with the instructor were significant characteristics of instructional delivery quality. These characteristics were confirmed by Eom and Ashill (2016), who used constructivist learning theory in a survey of 372 business students to examine the relationship of student perceptions of instructor involvement and facilitation, course design, satisfaction and learning outcomes. These findings are consistent with other research suggesting that instructor involvement and instructor-student and student-student interactions impact student satisfaction and learning outcomes in online formats (Garrison, 2016).

Our research builds on prior research by combining their findings on instructional characteristics to compare them across three modes of instructional delivery: in-person in-class (IC), online synchronous (SN) using live video conferencing and online asynchronous (AS) using recorded video. We separate online modes into synchronous and asynchronous because of the inherent difference in attention paid and responsiveness of students between these modes, and students' general preference for asynchronous instruction (Adkins & Tu, 2021). Along with the comparison of student perceptions, we also compare actual learning outcomes across the three modes of instructional delivery. The perceptions include instructor involvement, interaction amongst students, interaction with the instructor, course design and learning experience. Actual learning outcomes involve homework assignments and exams across in-class and online modes of instruction. In doing so, we extend the research conducted so far in a few unique ways. First, we compare student perceptions aggregated from several studies that relate to instructional delivery. Secondly, this multi-modal study is done within a single course, in a semester, and with one instructor, thereby reducing confounding effects when different student groups are subjects of the study. Thirdly, it examines perceived as well as actual learning outcomes.

3. METHODOLOGY AND ANALYSIS

Data for this study was collected from students in two sections of a core class in business analytics in a college of business at a Midwestern university. The course taught basic principles of descriptive, predictive, and prescriptive analytics using Microsoft Excel. The course was taught in each of the three modes of instruction, beginning with (i) in-person and in-class (IC), followed by online synchronous (SN) in which students received instruction using live video conferencing, and ending with (iii) online asynchronous (AS) in which students used video streamed on the Panopto™ platform till the end of the semester. In this manner, each student experienced all three modes of instruction.

A total of 61 students were surveyed for their perceptions of instructional characteristics. The survey was adapted from a study by Eom and Ashill (2016) who examined the determinants of student satisfaction and their perceived learning outcomes in the context of online learning. Items such as students' perception of instructor involvement (Ahmed, 2010) as well as learning items were added to the survey. Three attention checks were included in the survey to ensure that each respondent was paying close attention to the survey. The survey is included in Appendix A. After removing non-attentive responses, duplicates and incomplete responses, 48 data points remained for analysis.

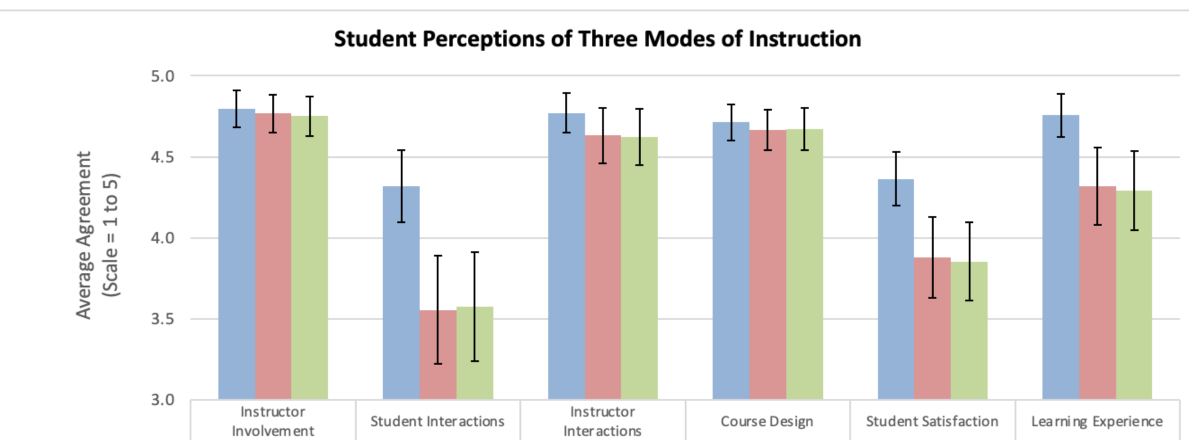
Data Analysis and Results

Since all students experienced each instructional mode, the study was appropriate for a repeated measures design. The survey examined levels of agreement across the six perceptions of instructional characteristics held by students and two learning outcomes. The six perceptions included instructor involvement, dialog amongst students, interaction amongst students and that with the instructor, course design, student satisfaction and learning experience. These perceptions were examined for each mode of instruction using a Likert scale from 1 to 5, with 1 representing strong disagreement and 5 representing strong agreement with the positive influence of each characteristic. The learning outcomes were collected at the end of each instructional mode.

The observations were sampled randomly and independently of each other. Academic performance was only compared across in-class and online modes (composite of asynchronous and synchronous modes) because the requirements of these modes were the same, i.e., this work was completed outside of class. A quick review of aggregate values for student perceptions (Figure 1) revealed that in general, they were highest for the in-class mode. Amongst the perceptions, student interaction, student satisfaction and learning experience appeared to drop more sharply for the two online modes than the other instructional factors. Quizzes and midterm exams were proctored in the same format and therefore aggregated as 'exam' at the conclusion of the in-class mode of instruction. Visual inspection of aggregate values for academic performance did not reveal strong differences between the in-class and online modes of instruction.

Figures 1 and 2 and the accompanying tables each show the comparison of averages, along with 95% confidence interval for these characteristics across the three modes of instruction. Not surprisingly, student perceptions of all six characteristics were lower for the modes of online instruction (SN & AS), than for in person (IC). Actual learning outcomes (Figure 2) were more mixed.

Internal consistency for all measures was tested using Cronbach Alpha and found to range between 8.0 and 9.5. The data violated assumptions of normality and homogeneity of variance. Therefore, we conducted the omnibus Friedman's test (Marino, 2018) with a repeated measures design for each construct using SPSS to find differences between the modes. Table 1 shows the output of the Friedman's test, which compares the mean rank for each characteristic across the three modes. This test outputs the results in the form of Chi square with p-values. Statistically significant differences between student perceptions are marked with an asterisk (*). Pairwise comparisons were conducted post-hoc using the Wilcoxon Signed Rank test to find the modes that differed (Table 2).



	Instructor Involvement	Student Interactions	Instructor Interactions	Course Design	Student Satisfaction	Learning Experience
In- Class	4.80	4.32	4.77	4.71	4.36	4.76
Synchronous Online	4.77	3.56	4.63	4.67	3.88	4.32
Asynchronous Online	4.75	3.58	4.63	4.67	3.85	4.29

Figure 3: Student Perceptions of Modes of Instruction

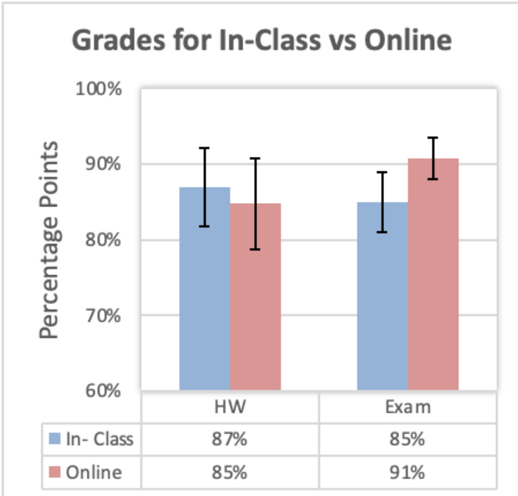


Figure 4: Grades for In class and Online Modes

Results in Table 1 and Table 2 indicate that the students perceived the in-class mode to have significantly better values than both online modes on all constructs except course design.

Actual learning outcomes were also examined pairwise across the modes of instruction using the Wilcoxon Signed rank test. As can be seen in Table 3, results were mixed. Although scores for the homework were higher for when the students were in-class, the difference between the in-class and online performance was not significant. However, students tended to do significantly better with online exams and tests than in-class.

Instructional Characteristics Means: Ranks→	IC	SN	AS	Chi-Square	df	Asymp. Sig.	Kendall's W
Instructor Involvement	2.17	1.95	1.89	7.585*	2	0.023	0.079
Interaction amongst Students	2.53	1.73	1.74	44.851*	2	0.000	0.467
Interaction with Instructor	2.22	1.92	1.86	13.216*	2	0.001	0.138
Course Design	2.11	1.92	1.97	3.959	2	0.138	0.041
Student Satisfaction	2.49	1.78	1.73	43.195*	2	0.000	0.450
Learning Outcomes	2.36	1.85	1.78	13.559*	2	0.000	0.329

** Significant at p < .05*
Table 5: Overall Test for Differences in Perceptions of Instructional Modes – Omnibus Friedman Test

	Instructional Characteristics											
	Instructor Involvement		Student Interaction		Instructor Interaction		Course Design		Student Satisfaction		Learning Outcomes	
Paired Comparisons	Z	* Asym. Sig. 2t	Z	Asym. Sig. 2t	Z	Asym. Sig. 2t	Z	Asym. Sig. 2t	Z	Asym. Sig. 2t	Z	Asym. Sig. 2t
SN-IC	-1.71[b]	0.088	-4.48[b]	0.000[a]	-2.42[b]	0.016[a]	-1.85[b]	0.06	-4.39[b]	0.000[a]	-3.74[b]	0.000[a]
AS-IC	-2.16[b]	0.031[a]	-4.46[b]	0.000[a]	-2.20[b]	0.028[a]	-2.00[b]	0.05	-4.13[b]	0.000[a]	-3.63[b]	0.000[a]
AS-SN	-1.19[b]	0.24	-.736[c]	0.461	-.11[b]	0.915	-.33[c]	0.74	-.71[b]	0.48	-.96[b]	0.336

** Asym. Sig. 2t represents Asymptotic Significance, two tailed*
a. significant at p < .05
b. Based on positive ranks
c. Based on negative ranks
Table 2: Pairwise Comparisons of Perceptions of Instructional Modes - Wilcoxon's Signed Rank Test

Paired Comparisons	Homework		Exams and Tests	
	Z	Asym.Sig. 2t *	Z	Asym.Sig. 2t *
Online vs In-class	-1.71[b]	0.088	-4.48[b]	0.000 [a]

** Asym. Sig. 2t represents Asymptotic Significance, two tailed*
a. significant at $p < .05$
b. Based on positive ranks
c. Based on negative ranks

Table 3: Analysis of Actual Learning Outcomes using Wilcoxon's Signed Rank Test

4. DISCUSSION

Student Perceptions of Instructional Characteristics

The descriptive information provided in figures 1 and 2 suggests students' preference for in-class instruction, across all characteristics: instructor involvement, interaction amongst students, interaction with instructor, course design, student satisfaction and learning outcomes. Analysis of data collected by the survey confirmed this for all characteristics but course design. Effect sizes (using Kendall's W) have been calculated for Friedman's test for each of these characteristics. The Kendall's W coefficient assumes a value from 0 (indicating no relationship) to 1 (indicating a perfect relationship). Kendall's W uses the Cohen's interpretation guidelines of 0.1 - < 0.3 (small effect), 0.3 - < 0.5 (moderate effect) and >= 0.5 (large effect).

1. Instructor involvement: this included providing timely feedback, encouragement, and facilitation of the course. Students perceived a difference in this characteristic when comparing in-class and asynchronous modes, but not with the synchronous mode of instruction. One reason for this could be that in both modes, the instructor is able to respond concurrently. In comparison, the asynchronous mode is perceived as being more latent, since it uses email or other non-current communication. This may have led to the perception of lowered involvement. Accordingly, the effect size of differences between the modes was found to be low (Kendall's W=0.079).

2. Interaction amongst Students: Students inherently interact with their peers when they are physically present. The ease of communication and interaction is clearly felt while comparing perceptions of interactions in-class with that in both synchronous and asynchronous modes of instruction. Peer-to-peer interaction has been recognized as having significant benefits, and its aspects have been well discussed (Pittman & Pike, 2016). In both online modes, this interaction is inherently reduced from that of in-class instruction, resulting in larger differences in student perceptions, with one of the largest effect sizes (W=0.467, medium-large).

3. Interaction with Instructor: Like the perceptions for dialog amongst students, students perceived that dialog with the instructor was significantly reduced during synchronous and asynchronous modes (W=0.138). While this may be true during asynchronous modes, it is interesting that students found the synchronous instruction to also have lower interaction than the in-class mode. One reason for this could be that students sense the absence of rich simultaneous in-person communication that takes place in-class.

4. Course Design: The lack of differences across the three modes of instruction delivery are not surprising, because course design was consistent across the three modes for all modules in the Business Analytics course (W=0.031). For each module, through the entire duration of the course across the three modes, students were first asked to follow step-by-step procedures shown in videos, to learn how to solve a set of problems. These videos demonstrated techniques and provided some theoretical background. For the second deliverable, students solved a sample problem live, with the instructor for the in-class mode as well as for the synchronous mode. For asynchronous instruction, this instruction also became a video that they needed to follow. The third and fourth deliverables for each module (i.e., homework and exams), had no change whatsoever, because students had to work on their

own and there was no instruction associated with those deliverables.

5. Student Satisfaction: This characteristic captured whether students liked working in this mode, such as doing presentations, taking quizzes, and learning from the instructor or other students. It exhibited some of the strongest differences between the in-class and both online instructional modes (W=0.450). In both the online modes, students had to depend on intrinsic motivation to pay attention to their work. We believe that there are a few reasons for this. While working asynchronously, the instructor is typically not available concurrently to support the student when they have a question. In the synchronous mode, only one student can be heard at a time. If a student seeks support for an issue, they may need to hold the entire class' attention to resolve a question – which can be a deterrent for introverted students.

6. Learning Outcomes: This characteristic captured students' perceptions about the quality of each mode, and whether it facilitated learning well. Students perceived strong differences between in-class and the two online modes of instruction (W=0.329). It is possible that some of this could be attributed to the fact that in the beginning of the course, students became accustomed to in-class instruction. In class, the instructor's presence motivated and compelled students to work on time. Switching to synchronous, and subsequently asynchronous modes, gradually put an increasing burden of timely work on the student, which required more intrinsic motivation.

Actual Learning Outcomes
Although it was clear that students preferred the in-class mode of instruction over the online modes, the learning outcomes did not clearly reflect improved performance with in-class instruction. Homework improved slightly for in-class but not significantly. However, performance on exams was significantly better for the online modes. One explanation for this is that in-class students are more aware of requirements of homework assignments due to richer in-class interactions. In contrast, online students must depend more heavily on intrinsic motivation. Homework carries less weight and therefore less importance in comparison with the exam. Therefore, homework may be less capable of drawing on intrinsic motivation and effort. As is often the case with analytics, answers can be completely correct or completely incorrect – i.e., there isn't always a middle ground. Students often under-estimated the time it would take them to complete homework correctly and before the deadline, incurring errors and penalties for late submissions. In comparison, exams carried a much more portentous appeal for preparation in advance, potentially causing more concern and driving the need to prepare better. It appears that for the online modes of instruction, students prioritized performance on exams to make up for lower performance regularity with course work (homework).

Student Commentary
The survey instrument collected open-ended comments from the students along the following lines:

1. Instructor Interaction*:* students appreciated quick responses to emails, flexibility, and personalized responses. They also acknowledged enthusiasm, positivity, and willingness to help with difficulties even when it took longer. This suggests that instructors should make a concerted effort to keep up interaction while switching to online modes.
2. Interaction with Other Students*:* Students reported that they often interacted with their peers to get support. Creating student groups was beneficial for students because it became a platform for them to interact with each other about issues, especially as interactivity was inherently reduced with online modes of instruction.
3. What Students Could Improve*:* Students acknowledged that they should attend more review sessions, be more proactive about reaching out to their own teams and use a central message/discussion board.

In general, the results show that students perceived in-class instruction to be most present and connected, followed by the synchronous mode of instruction. Even through course work such as reading or viewing videos for instruction is required, the student is not under direct supervision of the instructor during online modes. As a result, students only interact amongst themselves or with the instructor when necessary. Instructors may need to take this into account while working with online modes of instruction.

5. CONCLUSION, LIMITATIONS AND FUTURE RESEARCH

Analysis of data revealed that students preferred in-class instruction but tended to fare better on exams online. This research also suggests that instructors need to increase points of contact with students, create multiple check points, provide increased scaffolding, and perhaps create sliding scale for completing homework on time and with precision to reduce point loss for delays and inaccuracies. Students overwhelmingly acknowledged the appreciation and interest for interactivity. Instructors may facilitate interactivity by creating student groups to serve as a support system, and provision of other forms of scaffolding as appropriate.

One limitation is that there could have been some collaboration on online exams. Although this course leveraged special software to create individualized exam files for students and timed them carefully to minimize learning-on-the-fly, it may be difficult to completely rule out illicit collaboration. Students have often fared better in online assessments (Navarro & Shoemaker, 2000), but it is more common to see students faring better in-class (Sohn & Romal, 2015). Another possibility is that work at home may be less distracting, more comfortable, easing test anxiety and perhaps improves focus. A second limitation is that this research involved a business analytics class, which could limit its generalizability to classes that are similar. A third limitation is that this research had a small sample size,– a research constraint by way of having a single instructor and single course to ensure consistency of research. Since this study explores affects in the same course, it is possible that students became more comfortable with subsequent modes. Perhaps drawing samples from different sections for different modes could mitigate this, as well as the potential impact of any variation of complexity in course topics.

Further research may be needed to resolve the paradox of lower perceptions but better performance for online modes. Findings from this study could be corroborated with research using section-based separation of instruction modes, instead of using all modes in the same course.

The last two years have seen some flux in instructional design, wherein students and instructors alike moved to online instruction in combinations and variations such as HyFlex and blended instruction. In this state of flux, opinions and perceptions change as stakeholders of all types, from administrators to students, learn from their mistakes and improve on techniques. Therefore, additional research may also be needed to explore motivations, perceptions, and efficacies of various modes of instruction to stay abreast of this fast-changing nature of instructional delivery.

6. REFERENCES

Adkins, J. K., & Tu, C. (2021). Online teaching effectiveness: A case study of online 4-week classes in a graduate information systems program. *Information Systems Education Journal, 19*(3), 3.

Ahmed, H. M. S. (2010). Hybrid E-Learning acceptance model: Learner perceptions. *Decision Sciences Journal of Innovative Education, 8*(2), 313-346.

Ballenger, R. M., & Garvis, D. M. (2010). Student Usage of Instructional Technologies: Differences in Online Learning Styles. *Information Systems Education Journal, 8*(51), n51.

Beatty, B. (2014). Hybrid courses with flexible participation: The HyFlex course design. In *Practical applications and experiences in K-20 blended learning environments* (pp. 153-177): IGI Global.

Bowers, J., & Kumar, P. (2015). Students' perceptions of teaching and social presence: A comparative analysis of face-to-face and online learning environments. *International Journal of Web-Based Learning and Teaching Technologies (IJWLTT), 10*(1), 27-44.

Callister, R. R., & Love, M. S. (2016). A comparison of learning outcomes in skills-based courses: Online versus face-to-face formats. *Decision Sciences Journal of Innovative Education, 14*(2), 243-256.

Correia, A.-P., Liu, C., & Xu, F. (2020). Evaluating videoconferencing systems for the quality of the educational experience. *Distance Education, 41*(4), 429-452.

DiRienzo, C., & Lilly, G. (2014). Online versus face-to-face: Does delivery method matter for undergraduate business school learning? *Business Education & Accreditation, 6*(1), 1-11.

Eom, S. B., & Ashill, N. (2016). The determinants of students' perceived learning outcomes and satisfaction in university online education: An update. *Decision Sciences Journal of Innovative Education, 14*(2), 185-215.

Evans, N. S. (2013). *A cross-sectional descriptive study of graduate students' perceptions of* *learning effectiveness in face-to-face and online courses*: Wilmington University (Delaware).

Fadol, Y., Aldamen, H., & Saadullah, S. (2018). A comparative analysis of flipped, online and traditional teaching: A case of female Middle Eastern management students. *The International Journal of Management Education, 16*(2), 266-280.

Fish, L. A., & Snodgrass, C. R. (2015). Business student perceptions of online versus face-to-face education: Student characteristics. *Business Education Innovation Journal, 7*(2), 83-96.

Flanagan, J. (2012). Online versus face-to-face instruction: Analysis of gender and course format in undergraduate business statistics courses. *Academy of Business Research, 2*, 93-101.

Garrison, D. R. (2016). *E-learning in the 21st century: A community of inquiry framework for research and practice*: Taylor & Francis.

Hood, J., Jacques, L., Chen, Y., & Hebert, D. (2021). *Students' Perceptions on the Various Delivery Methods of Instruction.* Paper presented at the Society for Information Technology & Teacher Education International Conference.

Hrastinski, S. (2008). Asynchronous and synchronous e-learning. *Educause quarterly, 31*(4), 51-55.

Kakeshita, T. (2021). *Improved HyFlex Course Design Utilizing Live Online and On-demand Courses.* Paper presented at the CSEDU (2).

Kelly, R. (2021). 73 Percent of Students Prefer Some Courses Be Fully Online Post-Pandemic. *Campus Technology*.

Marino, M. J. (2018). Chapter 3 - Statistical Analysis in Preclinical Biomedical Research. In M. Williams, M. J. Curtis, & K. Mullane (Eds.), *Research in the Biomedical Sciences* (pp. 107-144): Academic Press.

Matta, V. (2021). Teaching in the online classroom: surviving and thriving in the new normal: by Doug Lemov and the Teach Like a Champion™ team, Published by Jossey Bass, A Wiley Brand, 192 pp., ISBN-13: 978-1119762935. *Journal of Information Technology Case and Application Research, 23*(1), 76-80.

Miranda, P., Isaias, P., Costa, C. J., & Pifano, S. (2017). Validation of an e-learning 3.0 critical success factors framework: A qualitative research. *Validation of an e-learning 3.0 critical success factors framework: a qualitative research*(1), 339-363.

Navarro, P., & Shoemaker, J. (2000). Performance and perceptions of distance learners in cyberspace. *American journal of distance education, 14*(2), 15-35.

Pathak, B. K., & Palvia, S. C. (2021). Taxonomy of higher education delivery modes: a conceptual framework. *Journal of Information Technology Case and Application Research, 23*(1), 36-45.

Pittman, J. M., & Pike, R. (2016). An observational study of peer learning for high school students at a cybersecurity camp. *Information Systems Education Journal, 14*(3), 4.

Raman, R., Sullivan, N., Zolbanin, H., Nittala, L., Hvalshagen, M., & Allen, R. (2021). Practical tips for HyFlex undergraduate teaching during a pandemic. *Communications of the Association for Information Systems, 48*(1), 28.

Sanford, D. (2017). Course format and learning: The moderating role of overall academic performance. *The International Journal of Management Education, 15*(3), 490-500.

Sebastianelli, R., Swift, C., & Tamimi, N. (2015). Factors affecting perceived learning, satisfaction, and quality in the online MBA: A structural equation modeling approach. *Journal of Education for Business, 90*(6), 296-305.

Sohn, K., & Romal, J. B. (2015). Meta-Analysis of Student Performance in Micro and Macro Economics: Online Vs. Face-To-Face Instruction. *Journal of Applied Business & Economics, 17*(2).

Sokout, H., Usagawa, T., & Mukhtar, S. (2020). Learning Analytics: Analyzing Various Aspects of Learners' Performance in Blended Courses. The Case of Kabul Polytechnic University, Afghanistan. *International Journal of Emerging Technologies in Learning (iJET), 15*(12), 168-190.

Weldy, T. G. (2018). Traditional, blended, or online: Business student preferences and experience with different course formats. *E-Journal of Business Education and Scholarship of Teaching, 12*(2), 55-62.

APPENDIX A

Survey items adapted from Eom & Ashill (2016), with Cronbach Alpha values for each mode.

#	Items	In-class	Asynchronous	Synchronous
Instructor Involvement		0.896	0.891	0.880
	The instructor was actively involved in facilitating learning.			
	The instructor provided timely helpful feedback on homework assignments.			
	The instructor provided timely helpful feedback on quizzes.			
	The instructor provided timely helpful feedback on student presentations.			
	The instructor provided timely helpful feedback on discussion forums.			
	The instructor stimulated students to exert intellectual effort.			
	The instructor cared about my individual learning.			
	The instructor was responsive to student concerns.			
Dialog amongst Students		0.838	0.923	0.914
	I had positive and constructive interactions with other students frequently.			
	The level of positive and constructive interactions among students was high.			
	I learned a lot from my fellow students.			
	The positive and constructive interactions among students helped me improve the quality of my learning outcomes.			
	What aspects of the student-to-student interaction impressed you the most to enjoy learning?			
	What could have helped you to improve student-to-student interactions in this mode?			
Dialogue with Instructor		0.829	0.833	0.853
	I had positive and constructive interactions with the instructor frequently.			
	The level of positive and constructive interactions between the instructor and students was high.			
	The positive and constructive interactions between the instructor and students helped me improve the quality of my learning outcomes.			
Course Design		0.783	0.811	0.787
	The course objectives and procedures were clearly communicated through the syllabus and explained in detail.			
	The course materials were interesting and stimulated my desire to learn.			
	The course materials supplied me with an effective range of challenges.			
	Student grading components such as homework assignments, presentations, quizzes, and exams were related to learning objectives of the class.			
Learning Experience		0.780	0.844	0.849
	The academic quality of this mode is excellent.			
	I have learned a lot from this mode.			
	The quality of the learning experience in this mode is great.			
Student Satisfaction		0.780	0.844	0.849
	I enjoyed doing presentations in this mode.			
	I enjoyed taking quizzes and tests in this mode.			
	I enjoyed learning in this mode from the instructor.			

	I enjoyed learning from peers in this mode.
Demographics	
	How old are you?
	What is your gender?
	What is your current year in school?
	What is your area of study?
	Before Spring Semester 2020, did you take an online course?
	If your answer was "Yes" to the previous question, was it fully online or blended?
Open Ended Comments	
	What aspects of the instructor impressed you the most?
	What could have the instructor done differently to make the learning environment even better?
	What aspects of your interaction with the instructor impressed you the most?
	What aspects of your interaction with other students impressed you the most to enjoy learning in the synchronous mode?
	What could have helped you to improve your interaction with the instructor?
	What could have helped you to improve your interaction with other students?

Coding Bootcamp Satisfaction: A Research Model and Survey Instrument

Guido Lang
guido.lang@quinnipiac.edu
Computer Information Systems
Quinnipiac University
Hamden, CT 06518, USA

Jason H. Sharp
jsharp@tarleton.edu
Marketing and Computer Information Systems
Tarleton State University
Stephenville, TX 76402, USA

Abstract

This study aims to shed light on what students like and dislike in coding bootcamps. A qualitative content analysis of student reviews for coding bootcamps was conducted, resulting in a research model and survey instrument consisting of fourteen factors that are proposed to affect coding bootcamp satisfaction. The proposed satisfaction factors include quality of instructors, value of mentors, availability of TAs, access to support staff, provision of career services, rigor of curriculum, appropriateness of pedagogy, development of peer connections, conduciveness of atmosphere, use of appropriate technology, affordability, openness of communication, quality of prep course, and level of post-bootcamp support. Each of the proposed satisfaction factors is measured with three to ten Likert-style variables. The proposed research model and survey instrument can be used by administrators and educators in coding bootcamps and traditional universities alike to better understand and ultimately improve student satisfaction in computing education.

Keywords: coding bootcamps, student satisfaction, student reviews, content analysis

1. INTRODUCTION

Commonly associated with military service preparation, the term bootcamp conjures up the image of intense, focused, and disciplined training of new recruits. Thus, the old adage, "no pain, no gain"! No longer just associated with military training, the bootcamp concept found its way into physical fitness in the late 1990s and gained recent popularity with CrossFit®, Fit Body®, and numerous others. Computing joined the movement as coding bootcamps began to first appear around 2011-2012 (Choxi, 2015; Waguespack, Babb & Yates, 2018). Subsequently, the number of coding bootcamps are on the rise worldwide, estimated by Course Report at over 500 (Course Report, 2021). A primary selling point of coding bootcamps is the cost and time savings over a traditional, four-year college degree (Waguespack, Babb, & Yates, 2018).

The purpose of this paper is to construct a research model and survey instrument for examining the factors behind coding bootcamp satisfaction. Given the sparse but growing amount of research related to coding bootcamps, there is little in the extant literature in the way of identifying what "students actually think of the [bootcamp] programs" (Bailey & Burke, 2019, p. 346). To this end, this paper addresses the following research question: *What are the factors driving coding bootcamp satisfaction?*

2. RELATED LITERATURE

With the growing interest and popularity of coding bootcamps around the world and their potential impact on computing education, there are several perspectives that arise in the research including industry, faculty, administrator, student, and curriculum (Burke & Bailey, 2019; Burke, Bailey, Lyon, & Green, 2018; Waguespack, Babb, & Yates, 2018).

Industry Perspective

Such questions as: "how do employers feel about hiring from four-year universities compared to coding bootcamps?" and "what types of skills are they looking for?" have been addressed (Burke et al., 2018). In relation to the first question, a good number of industry representatives indicated that a four-year degree is a requirement, however, not necessarily in a computing field. Others indicated that in some situations, they prefer hiring coding bootcamp graduates. It should be noted that 82% of coding bootcamp participants in their study already possessed a bachelor's degree (e.g., business, computer science, education, engineering, finance, liberal arts, communications, music theory) or higher, supporting their parallel work (Burke & Bailey, 2019).

For the second question, previous research indicates that "soft" skills were more prominently desired in the discussion with industry representatives (Burke & Bailey, 2020). These included skills such as teamwork, communication, along with ability and desire for continuous learning. While "hard" skills (e.g., programming) are a given, if an applicant is not able to get along with and work with others, the "hard" skills were found to be less important (Burke et al., 2018).

Faculty Perspective

Of the university faculty who participated in a related study (Burke et al., 2018), the consensus stated that their programs provided the necessary development of "hard" skills desired by industry representatives. However, in regard to "soft" skills, which were more highly discussed by the industry representatives, the faculty participants were mixed in their response about where and how these are developed in their curriculum. For a good number, skills such as teamwork, communication, and continuous learning are taught implicitly through specific assignments, team projects, and a capstone experience.

Administrator Perspective

While it is fairly common for academics to push back against the idea of training in higher education, among the coding bootcamp administrators and providers who participated in the study, they quickly recognized coding bootcamps as such and considered their "programs as experiential learning" (Burke et al., 2018, p. 506). With an emphasis on daily projects and assignments representing real-world problems and the workplace environment, administrators and providers felt they were providing students ample development in "hard" and "soft skills".

Student Perspective

The next perspective is that of the students who participated in a four-year college degree program in computing and those who participated in a coding bootcamp (Burke & Bailey, 2019). Results of the study indicated that for bootcamp students, getting a job during or shortly after completing the program was a primary focus. A large percentage, 86%, felt like hands-on project and peer collaboration was instrumental in learning and was implemented from the very start of the bootcamp. Other notable features of the learning environment included: industry partnerships, demo days, faced paced, innovation, immediate feedback, and a real-world work environment.

University students in contrast had not yet developed a clear plan for their careers. 86% reported a requirement to complete introductory coursework before moving on to advanced classes and completing capstone-type projects. There was less collaboration with industry compared to bootcamp students. The majority of university students reported learning communication and collaboration skills during the coursework, while 50% indicated development of other "soft" skills outside of the classroom. Across the four universities covered in the study, students reported the receipt of consistent feedback, but less immediate feedback when compared with bootcamp students. This was also true of industry collaboration, job acquisition, and practical, hands-on experience. All in all, bootcamp and university students showed very little difference in their perception of themselves as learners (Burke & Bailey, 2019).

Curriculum Perspective

The final perspective addressed in previous research involves the curriculum of coding bootcamps and four-year college degree programs in computing education. As noted by Waguespack, Babb, and Yates (2018), the majority of four-year college degree programs in computing education are accredited and guided by such organizations as AACSB or ABET. This requires these programs to meet certain standards of quality through assessment and continuous improvement. Coding bootcamps, on the other hand, are not regulated in the same manner. Despite their claims of cost and time savings, this often begs the question of the quality and oversight for coding bootcamps ("Are Bootcamps Booming?", 2016; Rafter, 2017). In an effort to place coding bootcamps within a context of comparison with four-year college degree programs in computing education, they triangulate coding bootcamps within the "curricular geography of CC2005" (p. 50). In doing so, they are able to map the competency target of coding bootcamps along the CC2005 field of computing competency continuum and then compare that mapping to the competency target of various information systems curriculum guidelines.

As the discussion of previous literature related to industry, faculty, administrator, student, and curriculum perspectives has shown, prior work has mostly focused on high-level comparisons between coding bootcamps and traditional university programs. The present work, in contrast, aims to shed light on the factors driving student satisfaction in coding bootcamps. The insights from this work should be able to contribute to the five perspectives mentioned earlier, while also holding implications for computing education more generally.

3. METHODOLOGY

To conduct this study, we collected data by scraping approximately 28,000 student reviews representing over 500 coding bootcamps from the Course Report website (n.d.). We then randomly ordered the student reviews to eliminate bias based upon type of bootcamp, location, length of review, or quality of review. To analyze the student reviews, we elected to use content analysis (Berg, 2001), a qualitative research technique.

Prior to starting the content analysis, we first established our process for evaluating each review to ensure consistency between us. We each then coded individually for a set time of 30 minutes by which to evaluate the process and the number of student reviews we were able to complete. After this initial round of analysis, we discussed any issues with the process and determined this was a reasonable approach for continuing.

As we analyzed the student reviews, we identified aspects of the coding bootcamp that students like and dislike. We continued this process individually until we each reached theoretical saturation. Theoretical saturation was reached when further analysis of student reviews revealed no further unique items for student likes and dislikes. We then began separately to group the "Like" and "Dislike" items into related categories. This led to the emergence of patterns and themes, which is the end result of content analysis. After working independently, we compared our results and began to further group the emerging themes and patterns in an attempt to cull down repeating ideas. After several iterations, we narrowed down the proposed factors driving bootcamp satisfaction for inclusion in our research model. Finally, we developed Likert-style items based on the identified variables. The full survey instrument based on the proposed research model is provided in Appendix A.

4. RESULTS

A total of fourteen satisfaction factors were identified. It is proposed that each factor positively influences coding bootcamp satisfaction. In turn, each factor consists of between three and ten variables. The following figure depicts the proposed research model, consisting of fourteen success factors and related propositions.

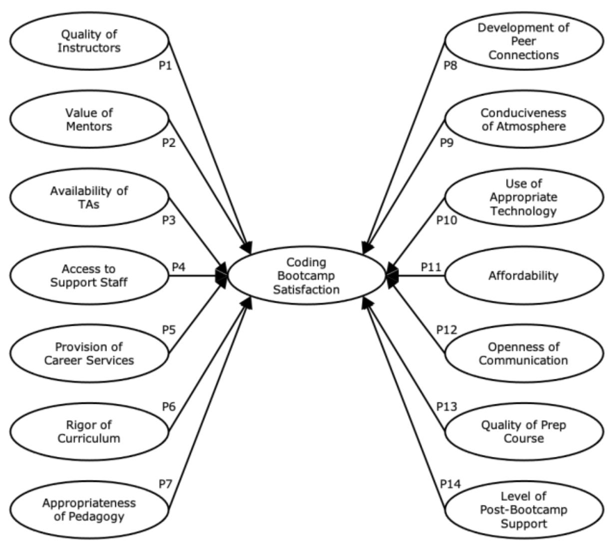


Figure 1: Proposed Research Model

The following sections describe the satisfaction factors along with their associated propositions.

P1: Quality of Instructors

The first proposed satisfaction factor is *quality of instructors*. The resulting proposition can be stated as: *the higher the quality of the instructors, the higher the coding bootcamp satisfaction* (P1). *Quality of instructors* consists of six variables, as shown in table 1.

ID	**The coding bootcamp ___________.**
QI1	Has instructors that are knowledgeable
QI2	Has instructors that are caring
QI3	Has instructors that are passionate
QI4	Has instructors with relevant industry experience
QI5	Has instructors that are inspiring
QI6	Has instructors that are available outside of class

Table 1: Variables Measuring Quality of Instructors

As indicated by the number of variables measuring *quality of instructors*, students appear to value different quality aspects in instructors. Among others, students appear to value the extent to which instructors are knowledgeable in the subject area (QI1). This is hardly surprising. However, other variables that emerged from the analysis are less obvious, such as the extent to which instructors are caring (QI2), passionate (QI3), and inspiring (QI5). This points to the importance of soft skills in instructors. Moreover, it is interesting to note that students appear to care about relevant industry experience (QI4). Lastly, students also wish for instructors to be available outside of class (QI6).

P2: Value of Mentors

ID	**The coding bootcamp ___________.**
VM1	Has mentors with relevant industry experience
VM2	Has mentors who are dedicated to students
VM3	Offers a variety of diverse mentors

Table 2: Variables Measuring Value of Mentors

The second proposed satisfaction factor is *value of mentors*. The resulting proposition can be stated as: *the higher the value of the mentors, the higher the coding bootcamp satisfaction* (P2). *Value of mentors* consists of three variables, as shown in table 2.

In the context of *value of mentors*, students appear to want to have mentors who are dedicated to the students' success (VM2), while also being offered a variety of diverse mentors (VM3). Here, diversity could refer to having mentors with a range of different social and ethnic backgrounds, genders, educational attainment levels, and professional experiences, etc. Moreover, students appear to be valuing industry experience in mentors (VM1), which underlines the primary value provided by mentors being in the area of career and personal coaching.

P3: Availability of TAs

The third proposed satisfaction factor is *availability of teaching assistants (TAs)*. The resulting proposition can be stated as: *the higher the availability of TAs, the higher the coding bootcamp satisfaction* (P3). *Availability of TAs* consists of three variables, as shown in table 3.

ID	**The coding bootcamp ___________.**
TA1	Has sufficient TAs available
TA2	Has TAs that are knowledgeable
TA3	Has TAs that are available outside of class

Table 3: Variables Measuring Availability of TAs

With regards to *availability of TAs*, students appear to place a special emphasis on the number of TAs available to them (TA1). In addition, students appear to value knowledge (TA2) and availability outside of class (TA3) in TAs. The latter two variables suggest that TAs play an important role in deepening the subject-matter understanding of students that should not be undervalued.

P4: Access to Support Staff

The fourth proposed satisfaction factor is *access to support staff*. The resulting proposition can be stated as: *the higher the access to support staff, the higher the coding bootcamp satisfaction* (P4). *Access to support staff* consists of three variables, as shown in table 4.

ID	The coding bootcamp ___________.
SS1	Has support staff that ensures students stay on track to graduation
SS2	Has support staff that helps students with administrative questions
SS3	Has support staff that is caring

Table 4: Variables Measuring Access to Support Staff

When it comes to *access to support staff*, students appear to value the help they can receive regarding staying on track to graduation (SS1) and regarding administrative issues involving the coursework and/or the bootcamp overall (SS2). Lastly, the extent to which support staff is caring towards students and their success in the bootcamp has been frequently mentioned by students (SS3). Thus, the role of support staff should not be solely focused on administrative efficiency but also have a strong personal support aspect.

P5: Provision of Career Services

The fifth proposed satisfaction factor is *provision of career services*. The resulting proposition can be stated as: *the higher the provision of career services, the higher the coding bootcamp satisfaction* (P5). *Provision of career services* consists of seven variables, as shown in table 5.

ID	The coding bootcamp ___________.
CS1	Helps find appropriate job openings
CS2	Prepares students for technical and non-technical interviews
CS3	Provides resume tips and reviews
CS4	Facilitates networking with industry professionals
CS5	Offers interesting company site visits
CS6	Hosts relevant guest speakers
CS7	Provides dedicated support for international job searches/applicants

Table 5: Variables Measuring Provision of Career Services

The *provision of career services* includes multiple aspects that are valued by students. Some of these aspects involve career services reaching out to industry, such as by facilitating networking with industry professionals (CS4), offering interesting company site visits (CS5), and hosting relevant guest speakers (CS6). Other aspects focus more on preparing students for the job search process, such as helping to find appropriate job openings (CS1), preparing students for technical and non-technical interviews (CS2), providing resume tips and reviews (CS3), and providing dedicated support for international job searches/applicants (CS7).

P6: Rigor of Curriculum

The sixth proposed satisfaction factor is *rigor of curriculum*. The resulting proposition can be stated as: *the higher the rigor of the curriculum, the higher the coding bootcamp satisfaction* (P6). *Rigor of curriculum* consists of six variables, as shown in table 6.

ID	The coding bootcamp ___________.
RC1	Teaches skills that are in demand
RC2	Teaches industry best practices
RC3	Gives a comprehensive introduction to a discipline
RC4	Provides an accelerated induction to a discipline
RC5	Balances soft and hard skills
RC6	Structures topics logically

Table 6: Variables Measuring Rigor of Curriculum

The *rigor of curriculum* in a coding bootcamp is determined by the curriculum's alignment with the needs of industry. This is reflected by the needs to teach skills that are in demand (RC1) and industry best practices (RC2). This requires coding bootcamps to maintain close industry contacts, to anticipate changes in industry demand, and to rapidly adjust their curriculum accordingly. In addition to teaching hard skills, students mentioned the importance of soft skills in a curriculum (RC5). As a whole, the curriculum should have enough breadth and depth to provide an introduction to a discipline that is both comprehensive (RC3) and accelerated (RC4), while progressing logically (RC6).

P7: Appropriateness of Pedagogy

The seventh proposed satisfaction factor is *appropriateness of pedagogy*. The resulting proposition can be stated as: *the higher the appropriateness of the pedagogy, the higher the coding bootcamp satisfaction* (P7). *Appropriateness of pedagogy* consists of ten variables, as shown in table 7.

ID	The coding bootcamp ___________.
AP1	Allows for learning at different speeds
AP2	Supports varying levels of prior knowledge
AP3	Balances conceptual and hands-on learning
AP4	Helps students become independent learners
AP5	Fosters collaboration among students
AP6	Challenges students without being overwhelming
AP7	Facilitates work on relevant, real-world exercises/projects
AP8	Incorporates appropriate assessments with timely and detailed feedback
AP9	Gives students individualized instruction
AP10	Encourages students to fully immerse themselves in a discipline

Table 7: Variables Measuring Appropriateness of Pedagogy

Clearly, *appropriateness of pedagogy* is an important and multi-faceted factor in determining coding bootcamp success. Some of the pedagogical aspects can be implemented through scaffolding, such as allowing for learning at different speeds (AP1) and supporting varying levels of prior knowledge (AP2). Moreover, students mentioned the need to balance conceptual and hands-on learning (AP3), fostering collaboration among students (AP5), facilitating work on relevant, real-world exercises/projects (AP7), and incorporating appropriate assessments with timely and detailed feedback (AP8). Other pedagogical aspects appear to be broader in scope than a single lesson, such as helping students become independent learners (AP4), challenging students without being overwhelming (AP6), and encouraging students to fully immerse themselves in a discipline (AP10). Lastly, students mentioned the wish for getting individualized instruction (AP9), which is a pedagogical aspect that could be implemented by changing the instructor-to-student ratio or leveraging adaptive learning technology, for example.

P8: Development of Peer Connections

The eighth proposed satisfaction factor is *development of peer connections*. The resulting proposition can be stated as: *the higher the development of peer connections, the higher the coding bootcamp satisfaction* (P8). *Development of peer connections* consists of three variables, as shown in table 8.

ID	The coding bootcamp ___________.
PC1	Ensures peers have comparable prerequisite knowledge and skills
PC2	Fosters social bonding among peers
PC3	Maintains appropriately sized cohorts

Table 8: Variables Measuring Development of Peer Connections

The *development of peer connections* factor aims to ensure that social bonding is supported among students (PC2) in an appropriately-sized cohort (PC3). The latter depends on the modality and facilities of the coding bootcamp, as there probably isn’t one cohort size that fits all coding bootcamps. While having a heterogeneous cohort in terms of background and experiences is probably beneficial, students specifically mentioned the desire for peers to have comparable prerequisite knowledge and skills (PC1), thus ensuring that peers will be able to collaborate well.

P9: Conduciveness of Atmosphere

The ninth proposed satisfaction factor is *conduciveness of atmosphere*. The resulting proposition can be stated as: *the higher the conduciveness of the atmosphere, the higher the coding bootcamp satisfaction* (P9). *Conduciveness of atmosphere* consists of three variables, as shown in table 9.

ID	The coding bootcamp ___________.
CA1	Maintains a positive and supportive atmosphere
CA2	Fosters a community feeling
CA3	Instills confidence and professionalism

Table 9: Variables Measuring Conduciveness of Atmosphere

Conduciveness of atmosphere is an interesting success factor that isn't easy to put into practice as it requires varying degrees of cooperation between staff and students. The aspect which probably requires the least amount of support from students is instilling confidence and professionalism (CA3). This aspect is solely the responsibility of the instructors and to a lesser extent the TAs and support staff. However, maintaining a positive and supportive atmosphere (CA1) along with fostering a community feeling (CA2) are both aspects that require both role-modeling from the entire staff along with cooperation from the students.

P10: Use of Appropriate Technology
The tenth proposed satisfaction factor is *use of appropriate technology*. The resulting proposition can be stated as: *the higher the use of appropriate technology, the higher the coding bootcamp satisfaction* (P10). *Use of appropriate technology* consists of five variables, as shown in table 10.

ID	The coding bootcamp ___________.
AT1	Supports collaboration among students with appropriate technology
AT2	Enables socialization among students via appropriate technology
AT3	Facilitates Q&A sessions and discussions using appropriate technology
AT4	Presents and shares learning materials through appropriate technology
AT5	Uses appropriate technology for assignment submissions and feedback

Table 10: Variables Measuring Use of Appropriate Technology

With regards to the *use of appropriate technology*, it appears that students desire appropriate technology for every aspect of their student experience. This includes technology used for learning in lessons, such as to present and share learning materials (AT4) and to facilitate question and answer sessions along with discussions (AT3). In addition, students look for appropriate technology to support them collaborating (AT1) and submitting assignments (incl. receiving feedback on assignments) (AT5). Lastly, students value having appropriate technology that enables them socializing within the cohort (AT2).

P11: Affordability
The eleventh proposed satisfaction factor is *affordability*. The resulting proposition can be stated as: *the higher the affordability, the higher the coding bootcamp satisfaction* (P11). *Affordability* consists of three variables, as shown in table 11.

ID	The coding bootcamp ___________.
AF1	Prices its offering competitively
AF2	Offers attractive tuition reimbursement options
AF3	Provides flexible tuition loan options

Table 11: Variables Measuring Affordability

Given the rising cost of higher education, the *affordability* of coding bootcamps is certainly a factor that is on students' minds. In this realm, students look for competitive pricing (AF1) along with flexible tuition loan options (AF3), the latter of which is typically provided by the bootcamp in collaboration with third-party financial organizations. The ability to receive tuition reimbursement after the start of a bootcamp (AF2) is another aspect that students look for when evaluating coding bootcamps.

P12: Openness of Communication
The twelfth proposed satisfaction factor is *openness of communication*. The resulting proposition can be stated as: *the higher the openness of communication, the higher the coding bootcamp satisfaction* (P12). *Openness of communication* consists of three variables, as shown in table 12.

ID	The coding bootcamp ___________.
OC1	Communicates openly and transparently with students
OC2	Regularly asks for students' feedback
OC3	Makes changes based on students' feedback

Table 12: Variables Measuring Openness of Communication

Openness of communication is a success factor that requires coding bootcamps to pay attention to the openness and transparency with which they communicate with students (OC1). Given the impact of COVID-19 on coding bootcamps, openness and transparency in communication was especially valued by students during that time. Moreover, students expect to be asked for feedback regularly (OC2) and for coding bootcamps to make appropriate changes based on their feedback (OC3). While the practice of asking for teaching evaluations is wide-spread in higher education, the desire of students to see the impact of their feedback is something that is frequently overlooked.

P13: Quality of Prep Course
The thirteenth proposed satisfaction factor is *quality of prep course*. The resulting proposition can be stated as: *the higher the quality of the preparatory course, the higher the coding bootcamp satisfaction* (P13). *Quality of prep course* consists of three variables, as shown in table 13.

ID	The coding bootcamp ___________.
QP1	Provides a thorough preparatory course
QP2	Has a preparatory course that is well-designed
QP3	Sets appropriate expectations with the preparatory course

Table 13: Variables Measuring Quality of Prep Course

The *quality of the preparatory course* takes place before the start of the bootcamp, but appears to be important for students' success. As such, students want a preparatory course that is thorough (QP1), well-designed (QP2), and sets appropriate expectations (QP3) for the remainder of the bootcamp.

P14: Level of Post-Bootcamp Support
The fourteenth proposed satisfaction factor is *level of post-bootcamp support*. The resulting proposition can be stated as: *the higher the level of post-bootcamp support, the higher the coding bootcamp satisfaction* (P14). *Level of post-bootcamp support* consists of three variables, as shown in table 14.

ID	The coding bootcamp ___________.
PS1	Offers ongoing career coaching after completing the bootcamp
PS2	Provides continuous skill development after completing the bootcamp
PS3	Fosters the development of alumni relationships after completing the bootcamp

Table 14: Variables Measuring Level of Post-Bootcamp Support

Given the existence of the *level of post-bootcamp support* factor, it appears that students view their learning experience in the bootcamp from the perspective of lifelong learning. As such, students value receiving ongoing career coaching (PS1), continuous skill development (PS2), and the development of alumni relationships (PS3) after the completion of the bootcamp.

5. DISCUSSION

The purpose of this study is to understand what students like and dislike about their experience in a coding bootcamp. As new entrants in the computing education space, coding bootcamps hold the potential to disrupt and improve the student experience in post-secondary education. Thus, the ultimate goal of this work is to provide insights about how to improve the student experience in coding bootcamps and in computing education more generally. To this end, a qualitative content analysis of student reviews for coding bootcamps was conducted, which led to the development of a research model consisting of fourteen satisfaction factors and an associated survey instrument (see Appendix A).

Some of the satisfaction factors are probably interrelated, such as the expected impact of *quality of the preparatory course* on the *development of peer connections* (by ensuring that peers have adequate prerequisite knowledge and skills). Another potential interrelation between satisfaction factors is the expected impact of *use of appropriate technology* on the *conduciveness of the atmosphere* (by ensuring that students are able to socialize remotely within the cohort).

Looking at the number of variables associated with each satisfaction factor, it appears that five satisfaction factors are particularly complex: *appropriateness of pedagogy* (10 variables), *provision of career services* (7 variables), *rigor of curriculum* (6 variables), *quality of instructors* (6 variables), and *use of appropriate technology* (5 variables). In fact, one could argue that pedagogy, career services, curriculum, instructors, and technology make up the core offering of a coding bootcamp. Thus, it is possible that these five satisfaction factors will show a particularly strong association with coding bootcamp satisfaction in future research.

Contributions
The present study makes contributions to each of the five perspectives mentioned in the literature review. Specifically, in terms of industry perspective, this study points to the need to provide well-rounded career services and valuable mentors in order to build pathways from coding bootcamps to industry. In terms of faculty perspective, which can be broadened to include all instruction-related matters, this study suggests that the quality of faculty, the availability of TAs, and the appropriateness of pedagogy play an important role in determining coding bootcamp satisfaction. In terms of the administrator perspective, there are several aspects that need to be paid close attention to, including ensuring affordability, access to support staff, use of appropriate technology, openness of communication, quality of the prep course, and the level of post-bootcamp support. The student perspective should include a focus on the development of peer connections along with creating a conducive atmosphere. Lastly, the curriculum perspective should be extended with the insights from the rigor of curriculum factor, which requires close interaction with industry.

Limitations
The study is not without limitations. First, while we believe theoretical saturation was reached during our analysis, considering the sheer number of student reviews in the dataset, there is a possibility that analyzing more student reviews could potentially reveal additional factors contributing to student satisfaction. Second, although the Course Report website (n.d.) includes student reviews from over 500 coding bootcamps, it was the only data source used for the study. It is possible that gathering data from other sources such as the SwitchUp website (n.d.), which also contains a large number of student reviews, might yield more satisfaction factors.

Future Research
As noted in the limitations, the data for the study derived from a single source. For future research, we plan to gather coding bootcamp student reviews from additional sources. One such source is SwitchUp (n.d.), which reports to have over 20,000 verified student reviews. As the purpose of the study is to understand the factors driving coding bootcamp satisfaction, future research should follow-up with a quantitative evaluation of the research model. As such, our future research agenda involves contacting coding bootcamps in order to conduct a survey among students and/or alumni using the proposed survey instrument (see Appendix A). This would allow us to test the proposed survey instrument as well as provide rich results for both academic purposes and to the coding bootcamp providers. A final area of future research we will investigate is how the identified satisfaction factors might apply to higher education degree programs in computing education more generally.

6. CONCLUSION

While there are those who have been predicting the eventual demise and extinction of coding bootcamps, the opposite seems to be the case, at least for the time being. Thus research aiming to better understand student satisfaction in coding bootcamps constitutes a timely and relevant endeavor. To this end, this study developed a research model and survey instrument consisting of fourteen satisfaction factors. Future research is needed to evaluate the statistical properties of the proposed survey instrument.

Like the military and fitness industries before them, the concept of the coding bootcamp with its intense focus on providing a relevant, up-to-date, real-world educational experience in a timely manner and at a reasonable cost is causing many to reconsider a traditional four-year university degree. And although participants of coding bootcamps commonly talk about the challenges and difficulties they encountered, a common theme is that it is worth it in the end and you get out of it what you put into it. As one bootcamp participant stated, "This is one of the most challenging and rewarding things I've ever done" (Course Report, n.d.).

7. REFERENCES

Are Bootcamps Booming? (2016, Nov/Dec). *BizEd*, 15(6), 15.

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

Burke, B., & Bailey, C. S. (2019). Camp or College? SIGCSE '19: Proceedings of the 50th ACM Technical Symposium on Computer Science Education, 345–350.

Burke, B., & Bailey, C. S. (2020). Becoming an 'Adaptive' Expert. *Communications of the ACM*, 63(8), 56-64.

Burke, B., Bailey, B., Lyon, L. A., & Green, E. (2018). Understanding the software development industry's perspective on coding boot camps versus traditional 4-year colleges. SIGCSE '18: Proceedings of the 49th ACM Technical Symposium on Computer Science Education. 503–508.

Choxi, R. (2015). Coding Bootcamps are Replacing Computer Science Degrees. *VentureBeat*. Retrieved June 3, 2021 from https://venturebeat.com/2015/11/08/coding-bootcamps-are-replacing-computer-science-degrees/

Course Report (n.d.). Retrieved June 3, 2021 from https://www.coursereport.com/

Rafter, M. V. (2017, May 25). Camp for Coders. *Computerworld*. Retrieved June 3, https://www.computerworld.com/article/3191988/is-a-coding-boot-camp-right-for-you.html

SwitchUp (n.d.). Retrieved June 7, 2021, https://www.switchup.org/

Waguespack, L., Babb, J. S., & Yates, D. J. (2018). Triangulating Coding Bootcamps in IS Education: Bootleg Education or Disruptive Innovation?. *Information Systems Education Journal*, 16(8), 48-58.

Appendix A: Proposed Survey Instrument

Dependent Variable: Coding Bootcamp Satisfaction
On a scale from 1 (very dissatisfied) to 5 (very satisfied), how satisfied are you with the coding bootcamp?

Independent Variables
On a scale from 1 (strongly disagree) to 5 (strongly agree), please indicate your agreement with the following statements:

The coding bootcamp __________.

Quality of Instructors (QI)
QI1 Has instructors that are knowledgeable
QI2 Has instructors that are caring
QI3 Has instructors that are passionate
QI4 Has instructors with relevant industry experience
QI5 Has instructors that are inspiring
QI6 Has instructors that are available outside of class

Value of Mentors (VM)
VM1 Has mentors with relevant industry experience
VM2 Has mentors who are dedicated to students
VM3 Offers a variety of diverse mentors

Availability of Teaching Assistants (TA)
TA1 Has sufficient TAs available
TA2 Has TAs that are knowledgeable
TA3 Has TAs that are available outside of class

Access to Support Staff (SS)
SS1 Has support staff that ensures students stay on track to graduation
SS2 Has support staff that helps students with administrative questions
SS3 Has support staff that is caring

Provision of Career Services (CS)
CS1 Helps find appropriate job openings
CS2 Prepares students for technical and non-technical interviews
CS3 Provides resume tips and reviews
CS4 Facilitates networking with industry professionals
CS5 Offers interesting company site visits
CS6 Hosts relevant guest speakers
CS7 Provides dedicated support for international job searches/applicants

Rigor of Curriculum (RC)
RC1 Teaches skills that are in demand
RC2 Teaches industry best practices
RC3 Gives a comprehensive introduction to a discipline
RC4 Provides an accelerated induction to a discipline
RC5 Balances soft and hard skills
RC6 Structures topics logically

Appropriateness of Pedagogy (AP)
AP1 Allows for learning at different speeds
AP2 Supports varying levels of prior knowledge
AP3 Balances conceptual and hands-on learning
AP4 Helps students become independent learners
AP5 Fosters collaboration among students

AP6 Challenges students without being overwhelming
AP7 Facilitates work on relevant, real-world exercises/projects
AP8 Incorporates appropriate assessments with timely and detailed feedback
AP9 Gives students individualized instruction
AP10 Encourages students to fully immerse themselves in a discipline

Development of Peer Connections (PC)
PC1 Ensures peers have comparable prerequisite knowledge and skills
PC2 Fosters social bonding among peers
PC3 Maintains appropriately sized cohorts

Conduciveness of Atmosphere (CA)
CA1 Maintains a positive and supportive atmosphere
CA2 Fosters a community feeling
CA3 Instills confidence and professionalism

Use of Appropriate Technology (AT)
AT1 Supports collaboration among students with appropriate technology
AT2 Enables socialization among students via appropriate technology
AT3 Facilitates Q&A sessions and discussions using appropriate technology
AT4 Presents and shares learning materials through appropriate technology
AT5 Uses appropriate technology for assignment submissions and feedback

Affordability (AF)
AF1 Prices its offering competitively
AF2 Offers attractive tuition reimbursement options
AF3 Provides flexible tuition loan options

Openness of Communication (OC)
OC1 Communicates openly and transparently with students
OC2 Regularly asks for students' feedback
OC3 Makes changes based on students' feedback

Quality of Preparatory Course (QP)
QP1 Provides a thorough preparatory course
QP2 Has a preparatory course that is well-designed
QP3 Sets appropriate expectations with the preparatory course

Level of Post-Bootcamp Support (PS)
PS1 Offers ongoing career coaching after completing the bootcamp
PS2 Provides continuous skill development after completing the bootcamp
PS3 Fosters the development of alumni relationships after completing the bootcamp