THE EFFECTS OF COMMUNITY PARTNERSHIPS ON K-12 COMPUTER SCIENCE EDUCATION

By

Aaron Reed, Ed.D.

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 4, 2019

Keywords: Computer science, K-12 education, K-12 teachers, ECS, curriculum, website design

Approved by

Dr. Jakita O. Thomas, Chair, Associate Professor, Computer Science and Software Engineering
Dr. Xiao Qin, Professor and Director, Computer Science and Software Engineering
Dr. Cheryl D. Seals, Associate Professor, Computer Science and Software Engineering
Dr. Dean Hendrix, Associate Professor, Computer Science and Software Engineering
Dr. Nancy Garcia, Associate Professor, Media Communication, West Texas A&M University

ABSTRACT

Technology is a critical component of the American economy. Analysts have claimed it is imperative that the United States maintain its status as a worldwide leader in technology or its economy, standard of living, and security are in jeopardy. Yet every year, more technology jobs are created than can be filled by qualified college graduates. A dramatic increase in computer science workers is needed to keep pace with employer needs.

Massive challenges face America's computer science education system. At the K-12 level, many teachers are underqualified to teach technology subjects. Many computer science teachers have backgrounds in non-technical subjects and are often forced to start teaching computer science to keep their jobs. Teachers reported being overwhelmed with student needs and they therefore lack the time required to fine-tune their course curriculum. Despite the wealth of resources that exist to help people learn computer science, teachers report very few resources exist that focus on how to effectively teach computer science. Additionally, computer science teachers in Utah describe existing computer science course curriculum as being outdated, unengaging, and uninteresting to students. Particularly, teachers described Utah's Exploring Computer Science (ECS) I course curriculum as boring and lackluster.

To help K-12 computer science teachers in Utah, we created a partnership with the State Board of Education, administrators and faculty from St. Owensby College, and Utah high school teachers. Through this partnership, we created a series of alternate lesson plans for Utah's ECS I course. These lesson plans focus on active learning and engagement of students through problems and projects. After the fall 2018 semester, we administered a survey to the 38 registered users of the portal, seeking feedback on the effectiveness of the portal and the curriculum enhancements. We conducted in-person interviews with six teachers to collect qualitative data regarding the curriculum and portal.

Feedback from teachers was extremely positive toward both the portal and the curriculum. Teachers found the portal helpful and easy to navigate and the curriculum engaging and interesting to students. Yet problems such as lack of knowledge-sharing, persistent computer science self-efficacy concerns, and the need for more in-depth training persist.

ACKNOWLEDGEMENTS

I first must thank the Lord God Almighty, without whom not only would this dissertation not be completed, but the very thought of it, and us, would be totally irrelevant. Thank you for providing me the drive and motivation, for providing me breath, life, physical and mental capabilities, and for being there throughout my life in good times and in bad.

This dissertation would also not have been possible if not for the patience and support demonstrated by Dr. Jakita O. Thomas, my dissertation committee chair. Thank you! Huge thanks as well to the rest of the committee: Dr. Xiao Qin, Dr. Cheryl D. Seals, Dr. Dean Hendrix, and Dr. Nancy Garcia. Special thanks as well to my University Reader, Dr. Melody Russell.

I likewise would never have completed this without the love and support of my beautiful wife Stacy and our five amazing children. I love each of you with all my heart. Standing by your husband/father while he completes one doctoral program is exasperating. Doing so as they complete two doctoral programs in five years is beyond the load any family member should bear. Thank you for your patience and tolerance through this academic journey.

Finally, I need to thank all those who helped in the K-12 partnership including Tim Clark, Ben Fletcher, Brian Roadifer, Larry Scoffield, Jed Bodily, Brandon Jacobson, Bill Keil, Lezlie Harper, Lance Peterson, Angela Nederhand, Nicole Reitz-Larsen, Kathy Bekkemellom, and a slew of other Utah K-12 teachers. Thank you!

To all who read this, know that with a cheerful mindset, rugged persistence, and faith in God, all things are possible. No matter how dark the night, when you arise each day with a renewed sense of unbridled optimism and relentless positivity, good will always follow.

TABLE OF CONTENTS

ABSTRACT ii

ACKNOWLEDGEMENTS iv

LIST OF TABLES xiii

LIST OF FIGURES xiv

CHAPTER I: INTRODUCTION 1

1.1 Computer Science Student Interest 3

1.2 Purpose of the Study 5

1.3 Assumptions 5

1.4 Research Questions 6

1.5 Significance of the Study 7

1.6 Summary 8

CHAPTER II: LITERATURE REVIEW 10

2.1 Historical Context 10

2.2 Calls for Science Education Reform in Modern Times 12

2.3 A Legitimate Threat to the Nation 14

2.4 STEM Career Opportunities 16

2.5 Computer Science: The ‘T’ in STEM 17

2.6 Computer Science Career Opportunities 18

2.7 The Computer Science Worker Shortage 23
2.8 Student Interest in Computer Science 25
2.9 High School Computer Science Curriculum 27
2.10 ECS Utah 27
2.11 Computer Science Faculty Resources 28
2.12 High School Computer Science Faculty Resources in Utah 31
2.13 The Need for Improved Online Resources 32
2.14 Computer Science Teacher Identity, Self-Efficacy, and Enthusiasm 33
2.15 Conclusion 35
CHAPTER III: METHODOLOGY 39
3.1 Research Questions 40
3.2 Building Community Collaboration 41
3.3 A Collaborative Curriculum Portal 42
3.4 St. Owensby Computer Science Curriculum Enhancements 46
3.5 Research Design 48
3.5.1 Establishing the Paradigm 48
3.5.2 Type of Study 50
3.5.3 Instrumentation 51
3.6 Participants and Sampling 51
3.7 Data Collection 52
3.8 Data Analysis 53
3.9 Procedures 56
3.9.1 Participant Selection 56

3.9.2 The Interview Process 57
3.10 Context and Limitations 59
3.10.1 Context of the Study 59
3.10.2 Researcher Positionality 60
3.10.3 Limitations of the Study 61
3.10.4 Assumptions 61
CHAPTER IV: ST. OWENSBY CURRICULUM ENHANCEMENTS 63
4.1 Context and Rationale 63
4.2 Design Methodology 66
4.3 Lesson Plan Format 68
4.4 Lesson Plan Summary 70
4.5 Summary 71
CHAPTER V: CURRICULUM PORTAL 72
5.1 Context and Rationale 72
5.2 Requirements and Constraints 73
5.3 Development Methodology 76
5.4 Initial System Design 77
5.5 Revised System Design 80
5.6 User Interface Revisions 81
5.7 Final User Interface 82
5.8 Summary 87
CHAPTER VI: RESULTS 89
6.1 Research Questions 90

6.2 Participants 91
6.3 Interview Participant Demographics 91
6.4 Difficulties Faced by Utah Computer Science Teachers 93
6.4.1 Many Computer Science Teachers Do Not Understand Computer Science 93
6.4.2 A Lack of Classroom-Ready Resources 96
6.4.3 Poor Collaboration among Computer Science Teachers 98
6.5 Impact of the Curriculum Portal 99
6.6 Impact of the St. Owensby Curriculum Enhancements 101
6.7 Curriculum: PacMan 107
6.8 Curriculum: Fortnite 110
6.9 Computer Science Teacher Identity 112
6.9.1 Computer Science Teacher Identity Concerns 112
6.9.2 Impact on Computer Science Teacher Identity of the Portal and the St. Owensby Curriculum 114
6.10 Computer Science Self-Efficacy 116
6.10.1 Computer Science Self-Efficacy Concerns 116
6.10.2 Impact on Computer Science Self-Efficacy of the Portal and the St. Owensby Curriculum 118
6.11 Enthusiasm and Motivation 119
6.11.1 Enthusiasm Concerns 120
6.11.2 Impact on Enthusiasm of the Portal and the St. Owensby Curriculum 121
6.12 Knowledge Transfer 122
6.12.1 Knowledge Transfer Concerns 123

6.12.2 Impact on Knowledge Transfer of the Portal 125

6.12.3 Reasons for Not Sharing Ideas on the Portal.. 127

6.12.4 Ideas for Improving Knowledge Transfer 129

6.13 Summary 130

CHAPTER VII: DISCUSSION 132

7.1 Discussion of Findings 133

7.1.1 RQ 1: Difficulties Faced by Utah K-12 Computer Science Teachers....................... 133

7.1.1.1 Lack of subject matter understanding. 133

7.1.1.2 Dearth of classroom-ready resources. 134

7.1.1.3 Limited availability of collaboration opportunities. 136

7.1.2 RQ 2: Impact of the Curriculum Portal 137

7.1.3 RQ 3: Impact of the St. Owensby Curriculum 139

7.1.3.1 Designed through collaborative partnership. 140

7.1.3.2 Built by teachers, for teachers. 140

7.1.3.3 Focus on active learning and current, relevant topics. 141

7.1.3.4 Adherence to a central standard. 142

7.1.4 RQ 4: Impact of Knowledge-Transfer through the Portal 143

7.1.4.1 Aversion to knowledge transfer. 144

7.1.4.2 Teachers default to conferences. 144

7.1.5 RQ 5: Impact of Community Partnerships 145

7.2 Next Steps and Recommendations 146

CHAPTER VIII: Conclusion 148

8.1 Implications 148

8.2 Future Research 150
8.3 Conclusion 151
REFERENCES 156
APPENDIX A – INTERVIEW TEMPLATE 165
APPENDIX B – CURRICULUM AND PORTAL SURVEY 167
APPENDIX C – EMAIL INVITATION FOR SURVEY 174
APPENDIX D – INFORMATION LETTER AND CONSENT FORM 175
APPENDIX E – EMAIL INVITATION FOR INTERVIEW 179
APPENDIX F – ST. OWENSBY CURRICULUM ENHANCEMENTS 180
Module 1: Days 1-2 180
Module 1: Days 3-4 181
Module 1: Days 5-7 185
Module 1: Days 8-9 188
Module 1: Day 10 191
Module 1: Days 11-14 194
Module 1: Days 15-16 198
Module 1: Days 17-19 200
Module 2: Days 1-2 202
Module 2: Day 3 204
Module 2: Days 4-6 204
Module 2: Days 7-9 209
Module 2: Days 10-12 209
Module 2: Days 13-14 212

Module 2: Days 15-16 212

Module 2: Day 17 213

Module 2: Days 18-21 213

APPENDIX G – COMPLETE SURVEY RESULTS 215

LIST OF TABLES

Table 1. STEM earnings by education level 16
Table 2. Themes and Sub-Themes Appearing in Interview Transcripts 55

LIST OF FIGURES

Figure 1. STEM Employment by type of STEM occupation (Fayer, Lacey, & Watson, 2017) .. 19
Figure 2. Employment change by type of STEM occupation 2009 to 2015................................ 21
Figure 3. Projected new jobs for types of STEM occupation, 2014-2024................................... 22
Figure 4. Growth in STEM bachelor's degrees awarded 2005-2015 25
Figure 5 –AngularJS Common MVC Architecture 77
Figure 6 – Initial Portal System Architecture 79
Figure 7 – Portal Database Diagram.. 80
Figure 8 – Revised System Architecture 81
Figure 9 – Portal Login/Registration Screen 83
Figure 10 – Courses Page of the Portal... 83
Figure 11 – Course Page of the Portal 84
Figure 12 – Lesson Page of the Portal 85
Figure 13 – Comments Section of the Lesson Page of the Portal.. 86
Figure 14 – Admin Section of the Portal 86
Figure 15 – Edit Lesson Page of the Portal... 87
Figure 16 – Survey Results – Does the Portal Facilitate Curriculum Understanding?............... 100
Figure 17 – Did the St. Owensby Curriculum Help Teachers Build Interest in CS? 105
Figure 18 – Did the St. Owensby Curriculum Help Teachers Engage Students?....................... 106
Figure 19 – Would Teachers Recommend the St. Owensby Curriculum to Others? 106
Figure 20 – Does the curriculum portal layout make lessons easier to understand? 138

CHAPTER I: INTRODUCTION

There is an underreported crisis happening in the United States of America. This crisis threatens the nation's economic standing in the world, the standard of living of its citizens, and its national security (Adkins, 2012; Napoleoni, 2010; U.S. Department of Education, 2013). The cause of the crisis: a massive, growing gap in the number of qualified computer science workers compared to those needed by the nation's employers (Kalil & Jahanian, 2013; National Center for Education Statistics, 2015). As a result, the nation's technology employers cannot find enough qualified workers to fill their open positions (Fisher, 2015). Science and technology jobs account for 50% of the nation's sustained economic growth, but only 6% of U.S. workers are employed in these crucial fields (Adkins, 2012; Noonan, 2017). As more and more companies compete for the same insufficient numbers of employees, they are forced to pay exorbitant salaries. Some companies in the San Francisco Bay Area are paying salaries exceeding $100,000 to new employees regardless of education level because of the lack of qualified candidates (Fisher, 2015).

One key factor in the computer science worker shortage is the relatively low numbers of computer science graduates compared to the number of open computer science-related positions. In 2015, there were over 527,000 open computer science jobs in America and there were fewer than 60,000 computer science graduates who entered the workforce (National Center for Education Statistics, 2015). That gap is expected to widen over the coming years as the computer science job market is expected to continue to grow at over 13% annually (Noonan, 2017).

In an article in *Fortune Magazine* entitled *Wanted: Highly skilled tech workers, $100,000 plus salary, no college required*, published in 2015, the author warned that the technology industry may be heading in a direction where college credentials are no longer needed, or even wanted, because of the need to get all available workers into the industry as quickly as possible (Fisher, 2015). What appeared at the time to be an improbable future may be becoming reality. The dearth of computer science workers has given rise to a new education industry, the technology boot camp (Duffner, 2018). These boot camps are typically offered by non-accredited education providers and serve as a means for career changers to get into the lucrative computer science industry as quickly as possible (Duffner, 2018). Boot camps typically consist of intense, immersive courses in which students learn to program in as little as three months (Duffner, 2018). Michael Girdley, CEO and cofounder of boot camp provider Codeup, explained that he “saw a huge broken situation” and that he “had a bunch of friends who ran companies and couldn’t find tech talent” (Duffner, 2018, p. 1). Adam Seligman, VP of Developer Relations at Salesforce.com claimed that “boot camps are bypassing the traditional university model because they allow anyone with any background to get into software development and get a great job” (Duffner, 2018, p. 1).

One area of concern related to technology boot camps is their ability to improve a person’s employability throughout their lifetime (Bailey, 2017). Without the deep, well-rounded education that a person is likely to receive in a traditional university setting, boot camp graduates may find themselves unemployable at some future point in their careers (Bailey, 2017). In a 2017 article entitled *Access, Power, and the Framework of a CS Education Ecosystem*, Bobb and

Brown described how these alternative forms of education are creating disadvantages for aspiring computer science students of color (Bobb & Brown, 2017). The authors correlated types of education such as certificates or training vs. undergraduate and graduate degrees with occupation levels such as innovation vs. maintenance (Bobb & Brown, 2017). The authors pointed out that Black and Hispanic students are far more likely to enroll in alternative forms of CS education including boot camps and certificate or training providers and that those types of education providers are typically limited to maintenance-level positions (Bobb & Brown, 2017). The authors coined the term *technology ghetto* to describe the resulting disproportionate number of students of color who find their careers limited to employment in the lowest paying and least influential sector of the technology market (Bobb & Brown, 2017). This technology ghetto concept underscores the importance of increasing interest in computer science as a major among the nation's high school population. In turn, this begs the question: How can we get more students to graduate in computer science from the nation's colleges and universities?

1.1 Computer Science Student Interest

The computer science dilemma has two core components: student interest and student retention. Although student retention in computer science is a critical piece of the puzzle, it was outside the scope of this study which instead focused on attracting more students to computer science majors.

A core premise in this study was that improving computer science education in the nation's K-12 schools can have a positive impact on college students' computer science interest levels. Yet again, in K-12 computer science classrooms, we find major challenges. Computer science high school teachers report having access to insufficient resources for training and idea-sharing CSTA, 2015). Many of those teachers are self-taught or lack the content knowledge needed to effectively teach computer science (Yadev, Gretter, Hambrusch, & Sands, 2016). As a result, computer science teachers report feeling overwhelmed and underprepared to teach difficult computer science topics (Yadev, et al., 2016).

Additionally, computer science teachers report difficulty with curriculum concepts and finding ways to engage students (Ni & Guzdial, 212; Yadev, et al., 2016). The Los Angeles Unified School District launched an initiative to develop engaging and inclusive computer science curriculum (Goode, Chapman, & Margolis, 2012). The resulting curriculum program, *Exploring Computer Science*, or ECS, had educated over 2,000 students by 2012 with largely positive results in terms of engagement and computer science interest (Goode, Chapman, & Margolis, 2012). ECS was quickly adopted by school districts nationwide and in 2008, the State of Utah became the first state to adopt the program statewide (Hu, Heiner, & McCarthy, 2016).

Utah's high schools represented their own challenges for administrators of the program. With a wide range of urban and rural districts, administrators of ECS Utah reported that teachers had difficulty traveling far distances for required training programs ((Hu, Heiner, & McCarthy, 2016). This further exacerbated the concerns that teachers do not possess adequate training to effectively instruct students in computer science concepts. Among their lessons learned from the initial launch of the program, administrators reported that an effective program such as ECS depends on collaboration, idea-sharing, and community partnerships (Goode, Chapman, & Margolis, 2012). Further complicating the issues, technology is a rapidly evolving field and the curriculum concepts should frequently evolve to keep pace. Without access to proper resources such as online collaboration tools for idea sharing in rural or remote areas, and without continual improvements to curriculum, teachers will struggle to be effective in K-12 computer science education.

1.2 Purpose of the Study

The purpose of this study was to evaluate the effectiveness of collaborative online tools, curriculum enhancements, and community partnerships in improving Utah K-12 computer science education. With the massive and growing gap between the number of annual computer science graduates and the number of open computer science jobs, improving interest in computer science as a major for the nation's college students is of paramount importance. By improving the effectiveness of computer science education in the nation's high schools, we may be able to positively impact the number of college students interested in majoring in computer science in the nation's colleges and universities.

1.3 Assumptions

The major assumption in this study is the premise that if high school computer science courses can be improved *and* if teachers can leverage an online platform to help them to gain content knowledge and better prepare to teach computer science courses without having to travel long distances, two things will happen. First, teachers will be more effective K-12 computer science teachers. Second, more students will exit high school with an interest in computer science as a college major choice and/or a career. Although this study depends heavily on this assumption, the assumption itself is not unfounded. In fact, administrators involved in the design and launch of ECS in Los Angeles found that the program positively impacted the computer science interests of the students. After taking ECS, the percentage of students who expressed an interest in continuing to study computer science increased from 17% to 43% (Goode, Chapman, & Margolis, 2012). The administrators also found that approximately 60% of the students reported an interest in working in the computer science field (Goode, Chapman, & Margolis, 2012).

Most of the research in K-12 computer science education revolves around ways to improve student learning. Although the researcher acknowledges the importance of student learning and the critical nature of continually improvement in that arena, this study did not focus on student learning. The researcher posits that understanding and improving the mindset, general preparedness, and self-efficacy of teachers is perhaps equally important as we strive to improve computer science education at the K-12 level. Hence, this research will focus on K-12 teachers rather than students.

1.4 Research Questions

The following research questions guided this study:

1. What are the difficulties faced by Utah K-12 computer science teachers and how can technology tools and community partnerships be leveraged to alleviate some of those difficulties?
2. How are computer science teaching identity, self-efficacy, and enthusiasm of teachers impacted by using a collaborative curriculum portal developed through community partnerships?
3. How are computer science teaching identity, self-efficacy, and enthusiasm of teachers impacted by curriculum enhancements developed through community partnerships?
4. How does knowledge transfer through an interactive curriculum portal impact computer science teacher identity, self-efficacy, and agency in the classroom?
5. What are the perceptions of Utah high school computer science teachers and other key stakeholders regarding the impact of community partnerships on high school computer science courses?

1.5 Significance of the Study

In 1983, a commission of 18 leaders in industry, education, and government authored a paper entitled *A Nation at Risk* (Gardner, 1983). The paper was a harsh denunciation of the then current state of American education. Specifically, the authors warned that American leadership in technology and science was being lost and that, if the slide continued, the nation would lose its competitive edge in the world (Gardner, 1983). Since that time, numerous calls for education reform have continued. Yet America's lead in key areas continues to dwindle. America currently ranks 38th in the world in mathematics proficiency and 24th in science (OECD, 2015). Whether American education is failing, or other nations are improving is irrelevant; the nation is losing its edge in science and technology. Only 20 years ago, 40% of the world's scientists lived in the United States (Adkins, 2012). Currently, the nation is home to only 15% of the world's scientists. Calls for education reform have come from the highest levels with both President Obama and President Trump addressing the need to improve technology education in their State of the Union addresses (Soergel, 2017; The White House, 2011).

The growing computer science worker gap is perhaps the biggest sign that the nation's educational system is flawed and/or failing. America's status as the worldwide leader in technological innovation is critically important to its economy (Adkins, 2012). If the nation's technology employers cannot find the qualified workers to fill their jobs, that leadership in technology is in jeopardy. This crisis threatens the nation's economic standing in the world, the standard of living of its citizens, and its national security (Adkins, 2012; Napoleoni, 2010; U.S. Department of Education, 2013). Hence, it is critical that we improve technology education in the United States and increase the number of students who are interested in pursuing computer science as a college major and/or career. This study is meant to provide additional insight into various ways we can positively impact high school computer science education.

Elements of the literature review focus on difficulties faced by K-12 computer science teachers as well as the urgency of improving computer science education in America. This study is aimed at identifying ways to improve K-12 computer science teacher identify, self-efficacy, and agency. In order to improve computer science education, an understanding of the challenges faced by these teachers and ways to assuage those challenges is of paramount importance.

1.6 Summary

This chapter illustrated the purpose of the study, the research questions, and the significance of the study. An introduction to the critical nature of the study was provided. A more in-depth literature review of the problems surrounding the computer science worker shortage is presented in Chapter II. The literature review also provides the user with more context surrounding the challenges in high school curriculum as well as those facing high school computer science teachers. A case for the tools in this study and the study itself will be given through the literature review.

Chapter III discusses the methodology of the study. Research questions are presented and the type of study, the instrumentation, participants and sampling methods are discussed. Details regarding the design of the curriculum portal and the curriculum enhancements are presented. The data collection techniques and data analysis protocols are illustrated. Finally, the context of the study and researcher are presented along with the limitations of the study and assumptions.

Chapter IV will provide a detailed review of the design of the curriculum portal. Elements of the development such as methodology, technologies, timelines, and architecture will be discussed.

Chapter V illustrates the curriculum enhancements. A review of each of the alternative lesson plans will be provided along with context regarding the rationale of the lesson plan and its intended results compared to its corresponding existing lesson plan.

In Chapter VI, the findings of the study will be presented. The emergent themes from the qualitative data will be presented. Thoughts and analysis related to the study will be discussed.

Finally, a discussion of the findings of the study will be presented in Chapter VII. The discussion will provide context to the study and discuss the relevance of the findings. Implications for K-12 and higher education will be presented. Recommendations will be presented as well as calls for future research. The paper will conclude with a summary of the problem, the tools and the study of the effectiveness of those tools, findings, implications, and recommendations.

CHAPTER II: LITERATURE REVIEW

The focus of this research is to further the work of research into computer science student recruitment and retention through supporting teachers through an online collaborative portal. This chapter provides a review of existing literature on computer science career opportunities, the computer science worker shortage, and computer science student retention concerns. To provide context for the review of the literature and the study itself, a brief historical overview of STEM and computer science in America will provided. The final portion of the chapter will focus on computer science high school curriculum and instruction and the associated current needs in both areas.

2.1 Historical Context

Relevant events that led to the current state of science-related education in American date back to the end of World War II (Napoleoni, 2010). Starting almost immediately after the end of the war, the United States of America and the Soviet Union commenced a period of positioning for global technology and military dominance (Napoleoni, 2010). Just over a decade after the end of World War II, the high-stakes game of one-upmanship between the world's two remaining superpowers evolved into a period known as the Space Race, in which, from 1955 to 1975, the two nations battled for supremacy in spaceflight capability (Napoleoni, 2010). The beginning of the Space Race age fueled fears of national security on both sides of the globe (Napoleoni, 2010). As concerns over national security escalated, Americans turned their attention to their educational system. Citizens and government officials feared that the nation's science-related education had become outdated and stagnant and that, without a substantial reformation of that system, the nation would be unable to compete with the Soviet Union (Harris and Miller, 2005). American concerns culminated in 1957 when the Soviets successfully launched the world's first satellite into space, Sputnik 1 (Harris and Miller, 2005).

In 1958, in an attempt to improve America's science-related educational system, the U.S. Congress passed the National Defense Education Act (NDEA) (Harris and Miller, 2005). The NDEA was the first act aimed at improving education in science in technology in America and signaled a new movement of deeper involvement in the nation's educational system by the federal government (Harris and Miller, 2005). The impact of the NDEA on the educational system was significant. In the decade following the signing of the NDEA, America rose to become the premier technological superpower in the world (Harris and Miller, 2005). The decade's success culminated in 1969 with the launch of Apollo 11 and its successful landing on the moon (Harris and Miller, 2005).

Despite the success of the NDEA and the rise of American technological might, America's science and technology education was not without its flaws. By the early 1980's, new fears had arisen among the American public that the nation's educational system was again in danger of lagging behind the rest of the world (Gardner, 1983; Harris and Miller, 2005). In 1981, a commission comprised of 18 industry, education, and government leaders was assembled to evaluate the state of the nation's educational system (Gardner, 1983). The commission found that the nation's SAT scores had dropped nearly 40 points from 1963 to 1980 (Gardner, 1983). Additionally, high school aptitude scores and achievement test results were significantly lower in 1980 compared to those in 1973 and 1977 (Gardner, 1983). In 1983, the commission published their findings in a report entitled *A Nation at Risk* in which they warned that if the current trends in American education were not reversed, the nation would lose its competitive edge in industry, commerce, and science and technological innovation (Gardner, 1983).

2.2 Calls for Science Education Reform in Modern Times

Three and a half decades after the publication of *A Nation at Risk*, the warnings of the commission are proving prophetic. The most recent results from the Program for International Student Assessment (PISA), one of the largest and widely-used international student assessments, lists the United States ranked 38th in mathematics proficiency out of the 71 participating nations (OECD, 2015). In science proficiency, America performs better, but still ranks behind 23 other nations (OECD, 2015).

The United States lags behind a significant portion of the world in various higher education metrics as well, particularly related to science and technology. According to the National Science Foundation's Science and Engineering Indicators for 2018, nearly half of all bachelor's degrees awarded in China, Japan, Iran, and Israel were in the fields of science and engineering compared to less than 40% in the United States. Between 2000 and 2014, the number of science and engineering degrees granted in Romania, Turkey, Germany, Taiwan, and China grew at a rate of approximately 200% whereas the number granted in the United States grew only marginally during that same period (NSF, 2018). A total of 52 nations annually grant a higher percentage of science and technology degrees than the United States of America (NSF, 2012).

The lower percentages of science and engineering degrees may contribute to a decline in the science, technology, engineering, and mathematics (STEM) labor force. Only 20 years ago, the United States was home to 40% of the world's scientists (Adkins, 2012). Today, only 15% of the world's scientists live in America (Adkins, 2012). Experts warn that this decline in the

STEM workforce endangers the nation's economy, standard of living, and national security (Adkins, 2012; U.S. Department of Education, 2013).

Since the turn of the century, various groups have issued calls for American education reform, specifically regarding STEM disciplines. In 2005, the National Academies warned that American leadership in STEM disciplines was rapidly eroding (Augustine, 2005). The Academies published *Rising Above the Gathering Storm*, in which they presented four areas for improvement in STEM education which included: improve K-12 math and science education; recommit the nation to a focus on STEM research; train, recruit, and retain a larger percentage of the world's scientists; and build the STEM economy to improve ongoing innovation in the country (Augustine, 2005).

President Obama launched an ambitious initiative in 2009 called *Educate to Innovate* which was hailed as a collaborative campaign between the private sector, the non-profit and research communities, and the federal government (Launches, 2009). The goal of *Educate to Innovate* was to improve the nation's worldwide standing in terms of student outcomes in math and science (Launches, 2009). Again, in 2013, the Obama administration attempted to sound the alarm regarding American STEM education with the publication of *STEM Attrition: College Students' Paths into and out of STEM* (U.S. Department of Education, 2013). In the publication, the President's Council of Advisors on Science and Technology urged K-12 schools, colleges, and universities to partner together and identify ways to increase the number of STEM graduates in American institutions (U.S. Department of Education, 2013). In 2016, President Obama unveiled his Computer Science for All initiative which again aimed to improve computer science education nationwide (Smith, 2016). The CS For All program called on states to implement a detailed five-year plan through which all students were to have access to computer science education (Smith, 2016).

In 2017, President Trump continued the calls for improvement in STEM education by launching an initiative in which $200 million in annual grants would be funneled to K-12 schools working to improve STEM education (Soergel, 2017). The criticality of improving the nation's performance in STEM education is underscored by the regularity of these calls for reform. The common theme among these calls is that the security, economic strength, and standard of living are in jeopardy if STEM education is not drastically improved (Augustine, 2005; Launches, 2009; U.S. Department of Education, 2013; Soergel, 2017).

2.3 A Legitimate Threat to the Nation

Industrial, academic, and economic leaders have warned that globalization has created an environment in which other nations are rapidly catching up, and even surpassing American technological might (Launches, 2009; U.S. Department of Education, 2013). In their publication, *Rising Above the Gathering Storm*, the National Academies posited that the worldwide leadership in economics and technology the United States has enjoyed since the end of World War II is at risk of being lost (Augustine, 2005). The authors pointed out the rapid growth in STEM education in developing nations and, when comparing those numbers to the somewhat stagnant growth in America, claimed that the nation's lead in science and technology may soon be lost and, if it is lost, it may never be recoverable (Augustine, 2005). China, for example, has ambitions of becoming the worldwide innovation leader by 2050 and has made a commitment to significantly improve STEM education in its higher education system (Han & Appelbaum, 2018). From 2005 to 2015, the number of universities in China nearly doubled, growing from 1,792 to 2,560 with 44% of the students in those universities graduating in science or engineering (Han & Appelbaum, 2018). China is just one example of the rapid changes being implemented by other nations throughout the world and without keeping pace with those changes, the United States is in jeopardy of being left behind.

During his first term in office, President Obama attempted to underscore the critical nature of STEM workers claiming that STEM jobs are invaluable in helping the nation maintain a leadership position in the world (The White House, 2011). The STEM workers in the United States are responsible for 50% of the nation's sustained economic growth (Adkins, 2012). Yet only 6% of U.S. workers are employed in these crucial STEM fields (Noonan, 2017). Overall, STEM jobs provide invaluable fuel to the nation's economy.

Perhaps the largest concern among America's STEM employers is the inability to find qualified workers. Nationwide, STEM employers report difficulty in finding qualified candidates to fill open positions (Noonan, 2017). This STEM worker shortage has been looming throughout the decade. In 2012, President Obama's Council of Advisors on Science and Technology issued a call to increase the number of STEM graduates from the nation's colleges and universities by one million through 2020 to fill the anticipated job openings (Holdren & Lander, 2012). Yet despite the attention on STEM education and the calls to increase the number of STEM-capable workers, the Department of Labor projected that 2.5 million STEM-related jobs would go unfilled in 2018 (Smith, 2017). With the department's estimate of $85,000 per year as the average pay for those jobs, the nation's workers will lose $200 billion in lost wages (Smith, 2017). The lack of wages affects American spending and, in turn, economic growth and stability and the lack of productive workers slow the nation's employers and their ability to contribute to the economy (Smith, 2017). The STEM job growth is expected to continue, magnifying the already dangerous STEM worker shortage. The Bureau of Labor

Statistics projects STEM job growth will continue to outpace non-STEM job growth by nearly 50% through 2024 (Noonan, 2017).

2.4 STEM Career Opportunities

The positive side of the STEM worker shortage for American STEM workers is the translation of the lack of available workers to sustained job security and opportunity for STEM-capable workers. The law of supply and demand pushes wages higher as the STEM worker shortage creates a competitive environment for the nation's STEM employers. As a result, STEM workers earned 29% more than their non-STEM counterparts in 2015, up from a 26% difference in 2010 (Noonan, 2017). According to the Bureau of Labor Statistics, 93 of the 100 occupations categorized by the bureau as STEM-related earn wages above the national average (Fayer, Lacey, & Watson, 2017). As seen in Table 1, STEM workers earn more than non-STEM workers at all levels of education, with the largest differences coming at lower levels of educational attainment (Noonan, 2017).

Table 1. STEM earnings by education level

	Average Hourly Wage		*Difference*	
	STEM	**non-STEM**	**STEM**	**non-STEM**
High school diploma or less	$27.53	$16.21	$11.32	70%
Some college	$30.79	$19.09	$11.70	61%
Bachelor's degree	$39.28	$28.34	$10.94	39%
Graduate degree	$45.37	$35.16	$10.21	29%

Source: Noonan, 2017

In addition to increased wages, the worker shortage translates to more job opportunities for STEM workers. Unemployment is also lower for STEM workers compared to those in non-STEM professions. The Bureau of Labor Statistics cite the unemployment rate of STEM workers at 2.5%, less than half that of non-STEM workers (5.5%) (Noonan, 2017). The STEM worker shortage not only poses a threat to our nation's economy, it also represents enormous opportunity for those seeking careers in STEM.

2.5 Computer Science: The 'T' in STEM

It will be helpful to the reader to discuss the separation between STEM and computer science before proceeding to the following sections. Computer science is a STEM discipline, represented by the 'T' in STEM, or technology. When academics or economists discuss computer science in the context of job or degree growth, it is often done so within the umbrella of STEM. However, for reasons that will be clear as the reader continues, it is often helpful to make a distinction between the two and to evaluate computer science in isolation of the other STEM disciplines. In many ways, computer science students behave differently from their STEM counterparts. Shaw and Barbuti (2010) found four key areas in which computer science students do not follow the behavioral patterns found in other STEM disciplines. Those are:

1. Computer science students who do not take STEM AP exams perform at the same levels as those who did take such exams. In every other STEM discipline, those who took AP exams outperformed those who did not.
2. Computer science students who leave the major and drop out of school have high self-efficacy, even equaling that of those who persist in the major. In every other STEM discipline, students who drop out have significantly lower self-efficacy than those who persist.
3. Computer science students who did not claim to be interested in pursuing a doctoral degree performed just as well as those who do claim such an interest. In

every other STEM discipline, students who are not interested in doctoral studies underperform compared to their peers.

4. Computer science students who struggled in math and science in high school performed as well as their peers who did not struggle in such classes. In every other STEM discipline, students who struggled in math and science in high school were not as effective as those who did well in those courses (Shaw & Barbuti, 2010).

The findings of Shaw and Barbuti (2010) perhaps warrant deeper research and discussion of computer science student behaviors outside of STEM.

Additionally, education and employment statistics related to computer science are often outliers when compared to those of other STEM disciplines (Bureau of Labor Statistics, 2014; National Center for Education Statistics, 2016). Because computer science students behave differently than their STEM counterparts and the statistics in computer science are outliers compared to those of other STEM disciplines, it can be helpful to evaluate computer science outside of the context of STEM. However, academic research on computer science-specific recruiting and retention is relatively sparse compared to that of STEM holistically. It may also be unwise to dismiss all STEM-related research as irrelevant to computer science despite the apparent differences in behavior and statistics. Hence, throughout this literature review, we will discuss computer science both within the context of STEM as well as in isolation.

2.6 Computer Science Career Opportunities

Although the career opportunities in STEM look promising, in many ways they paint an inaccurate picture. Many of the statistics related to STEM career opportunities are significantly skewed by the overwhelmingly positive statistics in computer science. The most recent data from the Bureau of Labor Statistics, for example, show that computer science-related occupations makeup 45% of all STEM employment (see Figure 1) (Fayer, Lacey, & Watson, 2017).

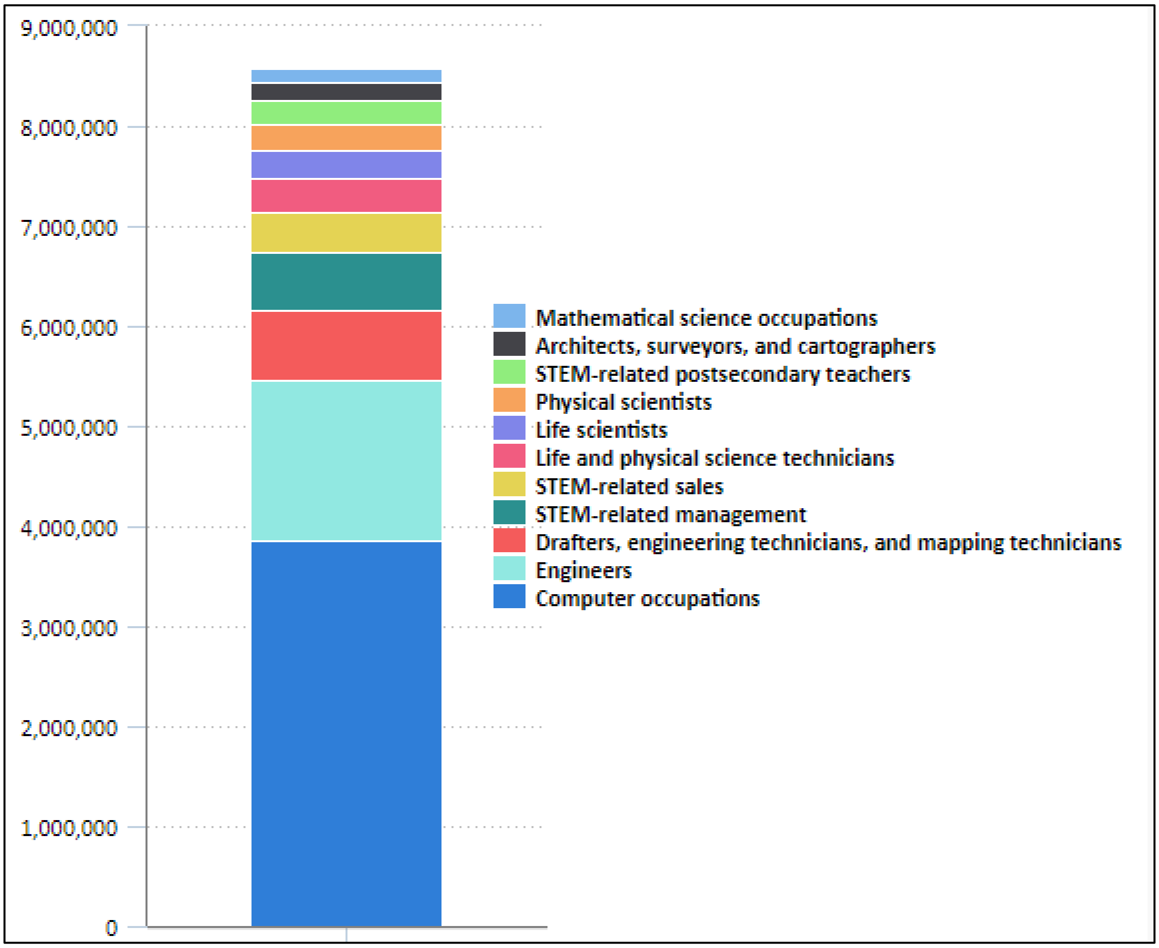


Figure 1. STEM Employment by type of STEM occupation (Fayer, Lacey, & Watson, 2017)

In all, seven of the top ten largest STEM occupations in 2015 were computer science-related. In fact, computer science occupations landed the top six spots and seven of the top eight with application software developers coming in at the top spot, computer user support specialists ranking second, computer systems analysts third, system software developers fourth, network and computer systems administrators fifth, computer and information systems managers sixth, and computer programmers landing at the eighth spot in the top ten list (Fayer, Lacey, & Watson, 2017).

Growth in STEM occupations has largely been computer science-related for the past decade. From 2009 to 2015, 80% of STEM job growth was attributable to computer science (Fayer, Lacey, & Watson, 2017). Computer science job growth outpaced every other STEM discipline by at least 500% (Fayer, Lacey, & Watson, 2017). In fact, if computer science-related job growth were to have been removed from the equation, STEM job growth from 2009 to 2015 would have dropped from an impressive 10.5% to a lackluster 2.1% growth rate (Bureau of Labor Statistics, 2017). Employment change data from 2009 to 2015 from the Bureau of Labor Statistics (2017) can be seen in Figure 2.

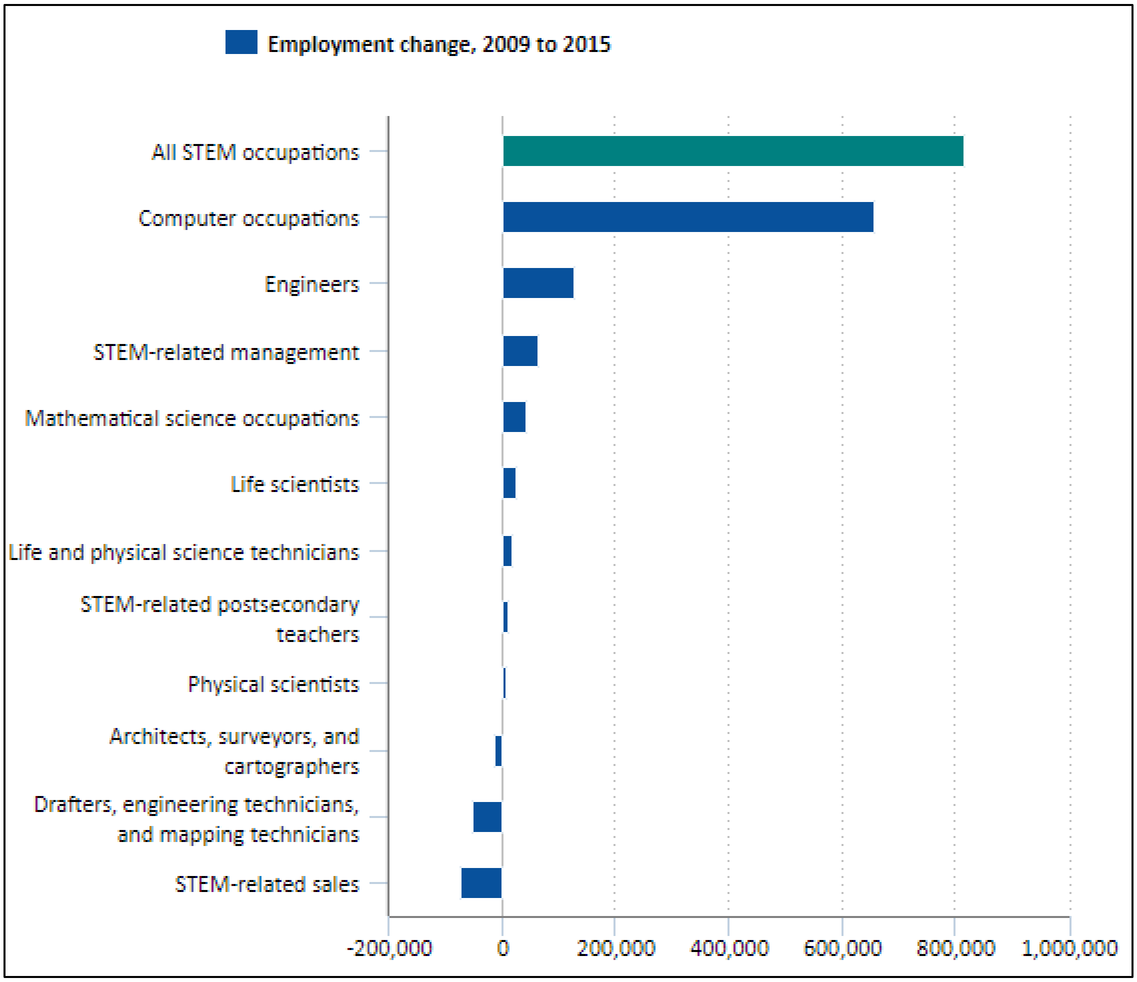


Figure 2. Employment change by type of STEM occupation 2009 to 2015 (Bureau of Labor Statistics, 2017)

Analysts predict that computer science's domination of STEM jobs will only continue to intensify. The Computing Research Association anticipates that through 2022, 65% of all STEM job growth will be in computing (Computing Research Association, 2014). In its estimates for job growth through 2024, the Bureau of Labors Statistics projects computer science occupations to grow at a rate of 13.1% while physical and life sciences are projected at 7.4% and engineering and architecture at 2.7% (Noonan, 2017). The number of projected new jobs in computer science-related positions is projected to be 7.5 times that of the next-fastest growing STEM field (engineering) and 1.6 times that of all other STEM fields combined (Fayer, Lacey, & Watson, 2017). The Bureau of Labor Statistics' projected job growth by STEM discipline can be seen in Figure 3.

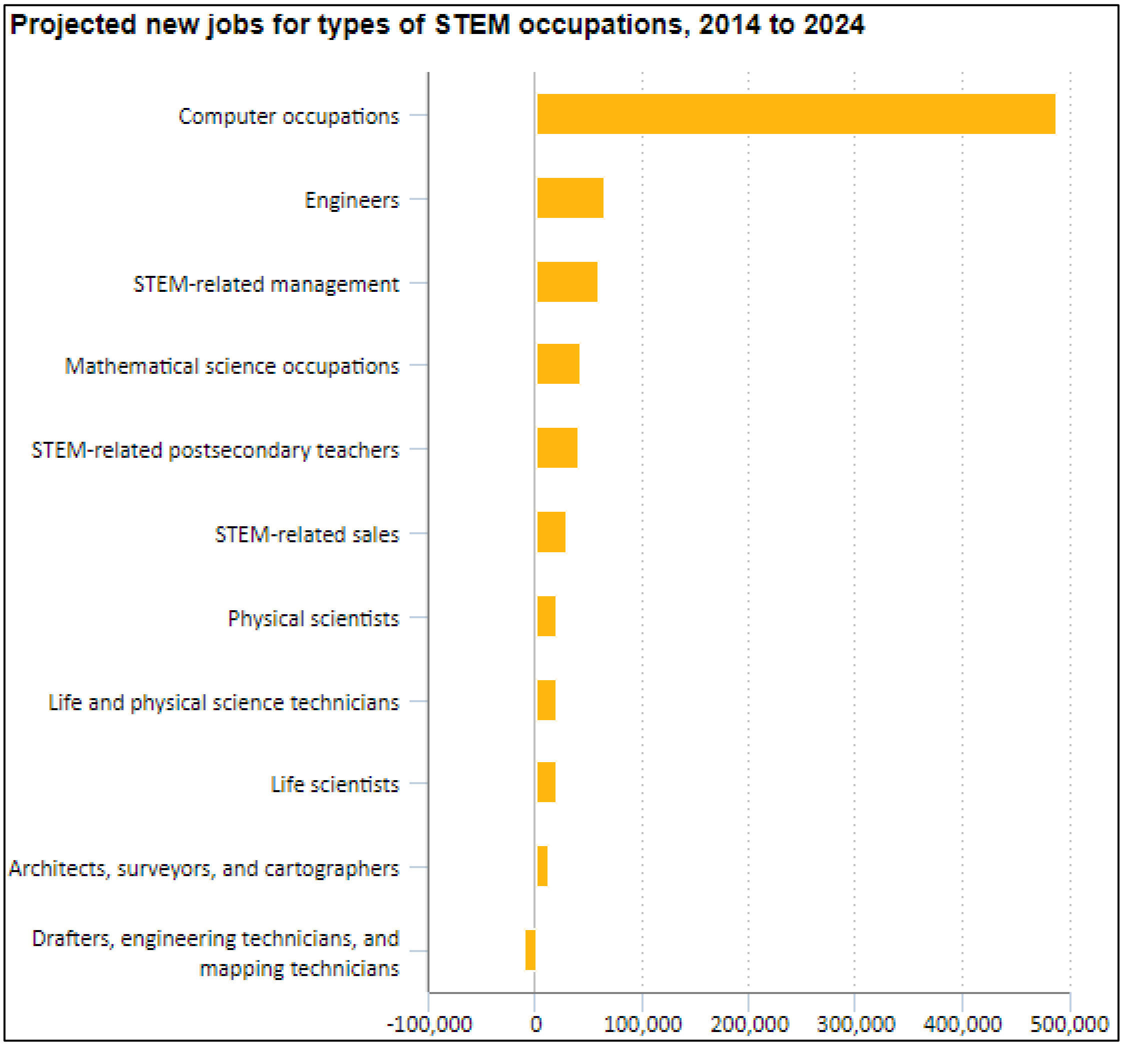


Figure 3. Projected new jobs for types of STEM occupation, 2014-2024 (Bureau of Labor Statistics, 2017)

Salaries for computer science positions are also stellar when compared to the rest of STEM. In its 2017 Salary Survey, the National Association of Colleges and Employers (NACE) projected that engineering students would be the top earners among STEM baccalaureate graduates at an average starting salary of $66,097 (NACE, 2017). NACE projected computer science students right behind engineering students at $65,540. According to NACE projections, engineering and computer science students would earn roughly 10% more than math and science majors. Computer science students were projected to earn higher starting salaries than all other STEM disciplines at the graduate degree levels. NACE projected computer science master's degree earners to have a starting salary of $81,039 and doctoral degree earners to have a starting salary of $110,841 (NACE, 2017). In comparison, engineering had the second-highest projected salaries at $75,053 for master's graduates and $95,973 for doctoral graduates (NACE, 2017). Not only do computer science careers represent a great opportunity for graduates in terms of job growth and available jobs, it also provides lucrative salaries compared to other STEM careers.

Particularly in terms of job growth, Shaw and Barbuti (2010) appear to be correct in their diagnosis that computer science does not necessarily follow the trends of the rest of the STEM disciplines. Whether the economists and analysts realize it or not, when they discuss positive momentum in STEM growth trends, they are largely talking about computer science first and all other STEM disciplines a distant second. For reasons that will be seen in the upcoming sections, the national conversation around STEM and STEM job growth should be modified to include at least some discussion about computer science in isolation from the rest of STEM to effectively address issues and concerns related to computer science, the fastest-growing discipline in STEM.

2.7 The Computer Science Worker Shortage

As computer science job growth projections continue to escalate, the nation's technology employers struggle to find qualified workers to fill those positions. In 2015, there were over 527,000 open computer science jobs and fewer than 60,000 computer science graduates to fill them (National Center for Education Statistics, 2015). The nation is on pace to have 1.4 million open computer science jobs by 2020 and only 400,000 available qualified candidates (Kalil & Jahanian, 2013). This shortage of available workers puts technology employers in competition with one another to hire the few qualified candidates. Forrester, one of the largest research and advisory firms in the world, predicted that in 2018, employers will be paying up to 20% above market rate salaries to attract the talent they need to sustain their enterprises (Forrester, 2018). Jonathan Godfrey, VP for Public Affairs at The App Association, a conglomerate of over 5,000 app developers and device manufacturers, claimed that "demand for developers is happening in every corner of this country, every region, every small town in America" (Rodriguez, 2018, p. 1). Alec Whitters, CEO of Higher Learning Technologies stated that his company is "working hard to get top technical talent, but there's just not enough out there for the demand seen in the country" (Rodriquez, 2018, p. 1).

One key component in the computer science worker shortage equation is the relatively low number of computer science graduates emerging from the nation's universities annually. In 2015, the number of computer science bachelor's degrees awarded in the United States was a meager 110% of the number that graduated in 2005 (National Center for Education Statistics, 2016). The number of bachelor's degrees awarded in each of the other STEM disciplines in 2015 was at least 150% higher than the corresponding 2005 number (National Center for Education Statistics, 2016). From 2005 through 2009, the number of annually awarded computer science degrees was in decline in America, compounding the worker shortage. Since 2009, the rate of growth in degrees awarded is on par with other STEM disciplines. Figure 4 shows the growth in degrees awarded from 2005 to 2015 for all STEM disciplines.

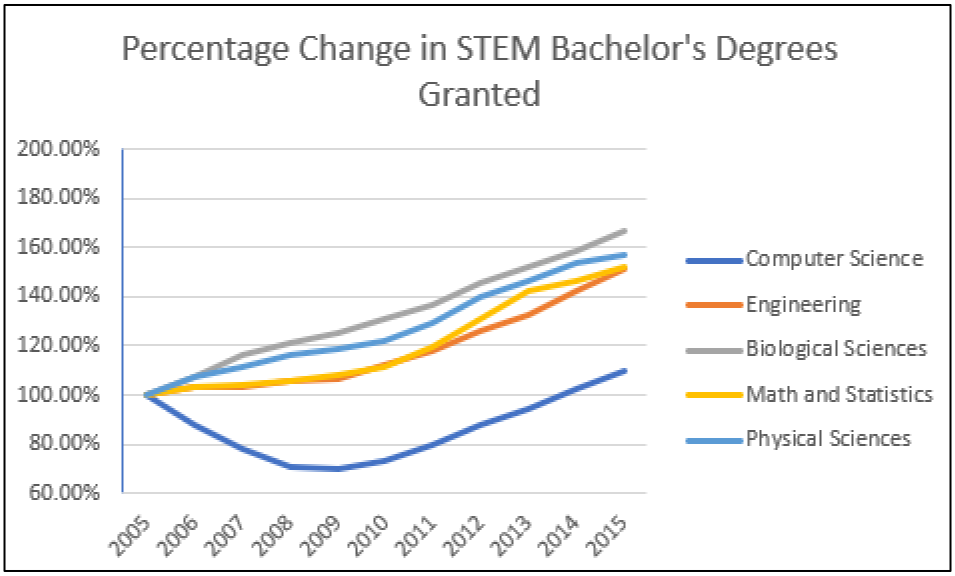


Figure 4. Growth in STEM bachelor's degrees awarded 2005-2015 (National Center for Education Statistics, 2016).

From 2005 to 2009, the number of bachelor's degrees awarded in computer science was in decline in America. Since 2009, the rate of growth is on par with other STEM disciplines (National Center for Education Statistics, 2016). However, because the job growth in computer science far outpaces the rest of STEM, the number of computer science degrees awarded must grow significantly faster if industry needs are to be met. Godfrey of The App Association claimed that "the simple truth is that we just don't produce enough software developers in this country" (Rodriguez, 2018, p. 1). He continues, stating that "it starts with education. We must start preparing students for [computer science] careers in primary and secondary schools."

2.8 Student Interest in Computer Science

With such promising career prospects and such low graduation numbers, outcome-minded administrators should be working to identify ways to increase their institution's output of computer science graduates. There are two primary methods to improve the computer science graduation numbers: increase the number of students who declare a computer science major and increase the graduation rate among computer science students. Although both are critical pieces of this puzzle, in this paper we will be focusing on increasing the number of students who declare a computer science major.

In 2006, Lori Carter of Point Loma Nazarene University conducted a study to identify why the number of students selecting computer science as a major was declining. At the time of her study, the number of students declaring computer science majors had declined by 60% compared to the previous four-year period (Carter, 2006). Carter surveyed 836 high school students about their interest in computer science majors in college. She focused on students who had taken calculus and pre-calculus in order to survey students who purportedly had the aptitude to succeed in computer science (Carter, 2006). In her results, Carter cited that 50% of the surveyed students cited a lack of understanding of computer science careers as a primary reason not to major in the field (Carter, 2006). Students claimed that their understanding of computer science careers was to sit at a computer all day long and not interact with others (Carter, 2006). Carter's study illustrated that students found computer science boring and lackluster and that they preferred to enter into fields which they found more interesting (Carter, 2006). Very few of the students in the study could articulate properly what computer science is and what students learn in the major.

Carter's research identified some core problems among American high school students: computer science has a reputation of being boring and students have little understanding of computer science as a major and a career (Carter, 2006). Others picked up on the problem as well as awareness grew regarding the deficiencies in computer science curriculum. In Computer Magazine, Qusay H. Mahmoud (2005) offered suggestions to improve computer science curriculum which included: (a) fixing computer science's image and helping students understand that there is more to the major and career than sitting at a computer all day, (b) make computer science courses more fun, engaging, and interesting, and (c) train the high school teachers to effectively deliver the computer science curriculum (Mahmoud, 2005).

2.9 High School Computer Science Curriculum

In response to the need for an enhanced computer science curriculum at the high school level, a robust initiative was launched in the Los Angeles Unified School District to create a new curriculum and program entitled *Exploring Computer Science* (Goode, Chapman, & Margolis, 2012). Created initially to address the issues of underrepresentation among minorities in computer science, Exploring Computer Science, or ECS, quickly evolved into a national program that was adopted by numerous school districts nationwide (Goode, Chapman, & Margolis, 2014). ECS was designed to introduce students to the concepts of problem solving and exposing them to computer science in practice, rather than rote syntax and programming (Goode, Chapman, & Margolis, 2014).

Initial results from ECS were positive and by 2012, over 2,000 students were enrolled in ECS courses (Goode, Chapman, & Margolis, 2012). After taking ECS, the number of students who self-reported interest in continuing to study computer science increased from 17% to 43% (Goode, Chapman, & Margolis, 2012). Additionally, approximately 60% of students reported an interest in working in the computer science field (Goode, Chapman, & Margolis, 2012).

2.10 ECS Utah

As the success of ECS became more widely publicized, the program continued to gain traction around the nation. In 2008, administrators in the State of Utah began a robust initiative in which ECS was adopted statewide (Hu, Heiner, & McCarthy, 2016). Led by Dr. Helen Hu, a computer science professor at Westminster College, the Utah Exploring Computer Science

Initiative began as a one-semester course as a partnership between Westminster College of Salt Lake City, Brigham Young University, and the Utah State Office of Education (ECS Utah, 2018). The ECS Utah program included curriculum adaptation and faculty development programs aimed to better prepare teachers for the ECS curriculum (ECS Utah, 2018). Between 2008 and 2012, over 150 Utah teachers were educated on the ECS curriculum and between 2012 and 2016, over 10,000 Utah students in 100 public and private schools participated in ECS courses (Hu, Heiner, & McCarthy, 2016). The Utah ECS initiative was the first statewide adoption of ECS (Hu, Heiner, & McCarthy, 2016). The initiative was somewhat unique in that it was one of the first programs to be offered in rural environments, with no additional cost for equipment, and in a half-year curriculum (Hu, Heiner, & McCarthy, 2016).

2.11 Computer Science Faculty Resources

The creators of the original ECS program in Los Angeles identified some critical lessons learned from the initial program offerings. One such lesson was a need to invest in local partnerships (Goode, Chapman, & Margolis, 2012). The authors found that such partnerships with professionals and educators can provide much needed resources and training for teachers and school systems at no up-front cost (Goode, Chapman, & Margolis, 2012). Additionally, the authors found that quality teachers are crucial in the deployment of an ECS curriculum (Goode, Chapman, & Margolis, 2012). Calling it the “elephant in the room,” the authors pointed out that without high-quality teachers, ECS is unlikely to be a success (Goode, Chapman, & Margolis, 2012).

In 2015, the Computer Science Teachers Association (CSTA) pointed out that the number of computer science-related teacher certificate programs is very limited (CSTA, 2015). Many of those programs that do exist teach very little related to how to teach computer science concepts and instead focus on the computer science skills themselves (Yadev, Gretter, Hambrusch, & Sands, 2016). The result is that teachers at the K-12 level are often teaching computer science with little or no training on computer science education (Yadev, et al., 2016). In a 2016 survey by CSTA, over 60% of responding teachers claimed that computer science is not a primary focus of their high school and is instead taught by business or mathematics teachers (Yadev, et al., 2016). In the same survey, 57% of teachers reported that computer science teachers also teach in other non-computer science areas (Yadev, et al., 2016). With a lack of training being compounded by an inability to focus solely on computer science, the teachers often lack the computer science content and pedagogical knowledge required to effectively educate their students (Yadev, et al., 2016).

In 2012, Ni and Guzdial from the School of Interactive Computing at Georgia Institute of Technology launched a study to identify characteristics of high school computer science teachers. The authors used semi-structured interviews to collect qualitative data on nine high school introductory computer science teachers (Ni & Guzdial, 2012). The study revealed that most of the teachers were either not fully committed to teaching computer science or were not confident in their ability to do so effectively (Ni & Guzdial, 2012). Several of the teachers were uncertain themselves regarding the validity of computer science as a field of study and a career option for their students (Ni & Guzdial, 2012). Finally, all of the teachers in the study felt isolated and unable to locate adequate sources for the training they required in computer science education (Ni & Guzdial, 2012). The findings of Ni and Guzdial underscore the difficult situation in which many computer science teachers find themselves as they seek to teach difficult concepts without proper training and access to resources. Unfortunately, this was also one of the primary lessons learned by the authors of ECS; faculty in ECS, and computer science in general, need access to resources and curriculum/training in order to adequately prepare to teach computer science concepts (Goode, Chapman, & Margolis, 2012).

Another study in 2015 by Menekse also illustrated the shortage of adequately prepared computer science teachers in America's high schools. According to the author, only about 10% of high schools nationwide have dedicated computer science instructors on staff (Menekse, 2015). What training is available for computer science teachers is often inadequate. Of the training programs identified by Menekse, only 33% lasted more than one week and the majority lasted less than one week (Menekse, 2015). The author reported that teachers felt inadequately prepared after leaving the training programs and indicated a desire to seek additional training elsewhere (Menekse, 2015). Nationwide, it appears that computer science teachers are lacking the critical curriculum materials and training resources they need to effectively teach computer science courses.

Computer science teachers who attend professional development often find themselves at a loss regarding how to implement what they learned in the classroom (Yadav, Gretter, & Hambrusch, 2015). Many computer science teachers claim they lack an adequate professional network of peers with whom to share ideas and best practices (Cutts, Robertson, Donaldson, & O'Donnell, 2017). Professional development sessions that are available are often not computer science-specific and, as a result, attendance in those sessions does not always translate to direct improvements in the classroom (Cutts, Robertson, Donaldson, & O'Donnell, 2017). This can result in an environment in which improvements in computer science teaching tactics are not effectively transferred to computer science teachers throughout the country (Cutts, Robertson, Donaldson, & O'Donnell, 2017). This environment where a significant number of computer science teachers are the only subject matter teachers at their school and they are limited in their knowledge transfer with their peers can stifle growth in terms of subject-specific pedagogy and curriculum development (Cutts, Robertson, Donaldson, & O'Donnell, 2017).

2.12 High School Computer Science Faculty Resources in Utah

Utah's ECS adaptation came with significant challenges. Faculty showed some resistance to the open-ended, exploratory nature of much of the ECS curriculum (Hu, Heiner, & McCarthy, 2016). The ECS curriculum contains probing questions meant to help students explore and feel comfortable answering ambiguous questions where there are no right or wrong answers (Goode, Chapman, & Margolis, 2012). These questions, such as "what is a computer?" and "what future problems could computers solve someday?" confused and frustrated some Utah teachers who felt they needed to know the correct answers and had a difficult time grasping the exploratory nature of the curriculum (Hu, Heiner, & McCarthy, 2016). Were some collaborative tools available for Utah teachers through which they could ask one another questions and cross train each other, these concerns may be alleviated.

Utah teachers voiced a preference to learn from mentors and training facilitators with Utah ties in their initial ECS training courses (Hu, Heiner, & McCarthy, 2016). The teachers portrayed a sentiment that those facilitators from Utah better understood the problems the teachers would face in the classroom, even more so than the creators of ECS from Los Angeles (Hu, Heiner, & McCarthy, 2016). This tendency may reinforce the notion that teachers have a desire for strong local and community partnerships through which they can learn and improve their teaching techniques together. The leaders of the Utah Exploring Computer Science Initiative also reported difficulties in recruiting teachers to attend workshops and training sessions, particularly those from more rural areas in the state (Hu, Heiner, & McCarthy, 2016).

This reinforces the need for online collaboration tools for faculty, eliminating, or at least reducing, the geographical concerns, in terms of travel, for training and mentoring.

2.13 The Need for Improved Online Resources

In 2015, Falkner and Vivian conducted a study in which they examined the effectiveness of online resources that were available for K-12 computer science teachers. Citing many of the previously mentioned concerns expressed by computer science teachers, the authors found a dearth of computer science-specific resources for K-12 teachers compared to those found in other subject areas (Falkner & Vivian, 2015). Although online resources that teach coding principles are plentiful, it is difficult and time consuming to find resources focused on computer science pedagogy and curriculum delivery methods (Falkner & Vivian, 2015). One of the biggest challenges for K-12 computer science teachers is determining whether or not a given academic resource is age-appropriate and fit for use in the classroom (Falkner & Vivian, 2015).

Gueudet and Trouche posit that when teachers utilize online resources, they often do not simply introduce them into their classrooms but instead teachers typically study the resources, combine them with other resources, revise them, and present them in their own ways (2015). This methodology of digesting and refining resources creates a sense of ownership of the course content and helps teachers more fully understand the materials themselves (Gueudet & Trouche, 2015). However, the lack of subject-matter expertise among most K-12 computer science faculty forces those teachers to seek out materials which are classroom-ready and can immediately be leveraged with their students (Gueudet & Trouche, 2015). Pedagogical resources for K-12 computer science teachers are scarce; the number of classroom-ready resources for a given age group, curriculum, and demographic is far smaller.

Perhaps the biggest area of weakness for teachers in the study by Gueudet and Trouche was the need to develop fresh ideas for student activities and projects (2015). Remillard and Bryans (2004) recommend that to be effective, teaching resources should be directed at teachers specifically, considering their needs and concerns rather than solely focusing on those of the students. Yet the vast majority of the resources evaluated by Gueudet and Trouche focus only on the student experience, dismissing, discounting, or simply ignoring the needs of the teacher (Gueudet & Trouche, 2015). To be effective at helping teachers improve in the classroom, online resources need to be teacher-focused, and ideally classroom-ready. Because it is difficult to create classroom-ready resources that will fit any classroom nationwide, regional resources that are refined for the classroom experience based on specific age groups, curriculum, and demographics may be more beneficial to teachers.

2.14 Computer Science Teacher Identity, Self-Efficacy, and Enthusiasm

Concerns expressed by Utah computer science teachers are not unique. Nationwide, computer science teachers struggle with a variety of efficacy-related issues (Cutts, Robertson, Donaldson, & O'Donnell, 2017; Diethelm, Hildebrandt, & Krekeler, 2009; Yadav, Gretter, & Hambrusch, 2015). Teacher identity is one such issue with which many high school computer science teachers struggle. Teacher identity can be thought of as the degree to which a teacher identifies themselves as a teacher of a particular subject (Ni & Guzdial, 2012). Teacher identity has been found to be a major factor directly related to teaching quality and teacher commitment (Cardelle-Elawar, Irwin, & Sanz de Acedo-Lizarraga, 2007; Chan, Lau, Nie, Lim, & Hogan, 2008). Ni and Guzdial (2012) found that a large percentage of high school computer science teachers did not identify as computer science teachers but instead thought of themselves primarily as teachers of other subjects. Computer science teachers who struggled with their identity demonstrated low motivation and low commitment to teaching technology-related topics (Ni & Guzdial, 2012).

Another factor impacting computer science teacher effectiveness is a lack of confidence in their ability to teach computer science topics (Cutts, Robertson, Donaldson, & O'Donnell, 2017). Many computer science teachers are experts in other subjects such as mathematics or business and learned computer science at the request of their administrators (Cutts, Robertson, Donaldson, & O'Donnell, 2017). Many others are self-taught or have learned the technical subjects alongside their students in class (Cutts, Robertson, Donaldson, & O'Donnell, 2017). The resulting low confidence among teachers negatively impacts their ability to effectively instruct their students. (Cutts, Robertson, Donaldson, & O'Donnell, 2017). Self-efficacy can be directly related to teacher confidence and reflects the beliefs teachers have about their own abilities to help students engage with the course material and meet the learning outcomes of the class (Woolfolk, Hoy, & Spero, 2005). In 2009, Diethelm, Hildebrandt, and Krekeler found that many computer science teachers have low self-efficacy in relation to their ability to teach computer science concepts. Low self-efficacy among computer science teachers has been linked to the constantly changing world of computer science, which results in a wide range of learning goals and outcomes which should be continually adapting with the technology industry (Diethelm, Hildebrandt, & Krekeler, 2009). Computer science teachers at the K-12 level often find themselves as the sole technology teacher at their school which limits their ability to collaborate, share ideas, and learn from their peers (Diethelm, Hildebrandt, & Krekeler, 2009; Yadav, Gretter, & Hambrusch, 2015). The inability to collaborate and share ideas with peers further exacerbates the self-efficacy issues many teachers experience. Many computer science teachers find themselves unpleased with their own class preparation and teaching and, without peers to help them improve, they find themselves struggling to build confidence in their own technology teaching skills (Bender, Schaper, Caspersen, Margaritis, & Hubwieser, 2016).

Another factor that directly impacts a teacher's efficacy is their intrinsic motivation. Eccles and Wigfield (2002) found a strong correlation between intrinsic motivation of teachers and students' ability to adapt and learn. Teacher's intrinsic motivation is closely related to teacher enthusiasm and can be measured by the degree of positive experience teachers encounter while teaching a given topic (Kunter, Frenzel, Nagy, Baumert, & Pekrun, 2011). Long and Hoy (2006) found that teacher enthusiasm is highly correlated with effective instruction and can directly impact the motivation of students. Factors which lead to low self-efficacy among computer science teachers including the lack of peer collaboration and the lack of subject matter expertise also negatively impact teacher enthusiasm and intrinsic motivation (Bender, Schaper, Caspersen, Margaritis, & Hubwieser, 2016).

K-12 computer science teachers who have found ways to connect with and collaborate with peers teaching similar subjects have found those professional networks to greatly improve their confidence, self-efficacy, and enthusiasm in and for computer science (Ni, Guzdial, Tew, Morrison, & Galanos, 2011). Many of those teachers who find themselves without such networks were found to be looking for more opportunities to collaborate in communities of computer science teachers (Ni & Guzdial, 2012). If computer science teaching is to be improved, it is critical that teachers are provided more opportunities to collaborate with their peers and learn from one another.

2.15 Conclusion

Ever since the end of World War II, the United States has been in a battle for technological and scientific global supremacy (Napoleoni, 2010). In the decades following the war, the primary opposition was the Soviet Union, as the two nations fought as the world's two premier superpowers (Napoleoni, 2010). In the heat of that fight, in 1958, the United States Congress passed the National Defense Education Act (NDEA), the first act directed specifically at improving the nation's science and technology education (Harris and Miller, 2005). Since the signing of the NDEA, the nation has seen repeated cries for reform of its science and technology educational system (Gardner, 1983; Harris and Miller, 2005). As America's educational system falls behind that of other nations, the country's status as the worldwide leader in technology is in jeopardy.

With technology driving much of the nation's economy, careers in science, technology, engineering, and mathematics (STEM) are on the rise (Noonan, 2017). However, a looming STEM worker shortage further threatens the nation's economy (Holdren & Lander, 2012; Noonan, 2017). With an estimated one million additional STEM workers needed to fill the open jobs, a nationwide emphasis on improving STEM education has begun (Fayer, Lacey, & Watson, 2017).

However, within STEM, one discipline seems to stand out from the rest in both employment and education statistics. Computer science-related positions makeup 45% of all STEM employment (Fayer, Lacey, & Watson, 2017). From 2009 to 2015, 80% of STEM job growth was computer science related (Fayer, Lacey, & Watson, 2017). Without taking computer science into consideration, the growth rate of STEM jobs falls from 10.5% to 2.1% (Bureau of Labor Statistics, 2017). When talking about the STEM worker shortage and the importance of enhancing STEM education, the bulk of the nation's efforts should be squarely directed at computer science education.

With such impressive growth of computer science jobs, the nation's universities are not able to keep up the pace to provide an adequate supply of laborers (Rodriguez, 2018). In terms of student recruitment into computer science majors, the nation's high schools require an improved and enhanced computer science curriculum (Goode, Chapman, & Margolis, 2014). A computer science curriculum program called *Exploring Computer Science* (ECS) was developed in 2008 and quickly became adopted by school districts nationwide (Goode, Chapman, & Margolis, 2014). However, the curriculum is nearly a decade old and students do not find the contents of the program as engaging and interesting as they used to, according to Jed Bodily, Manager of High School Relationships at St. Owensby College (personal communication, September 1, 2018). Enhancements to ECS are needed to continue the positive momentum generated by the curriculum program.

Additionally, computer science teachers nationwide find themselves unprepared for the subject matter both in terms of content and pedagogy (Yadev, Gretter, Hambrusch, & Sands, 2016). Teachers are often spread across other subjects in addition to computer science and typically lack expertise in computer science (Yadev, Gretter, Hambrusch, & Sands, 2016). A common thread among computer science teachers is a lack of resources and access to updated curriculum and training (Yadev, Gretter, Hambrusch, & Sands, 2016).

An acute need for updated curriculum which reenergizes the ECS program and helps teachers engage students has arisen. In conjunction with that need, computer science teachers require access to updates to the curriculum and areas for collaboration with peers and partnerships with industry or higher education providers to better prepare themselves for teaching in computer science. Without these enhancements, computer science teachers may be unable to provide students with the proper introduction to computer science which students need to determine their interest in computer science majors and careers (Goode, Chapman, & Margolis, 2014).

Finally, many computer science teachers struggle with their computer science teacher identity, instead identifying primarily as teachers of other subjects. Computer science teachers often have low computer science self-efficacy due to a lack of confidence in their ability to teach technical subjects. Computer science teachers also frequently struggle with intrinsic motivation and enthusiasm because of their lack of confidence and non-computer science identities. If administrators hope to improve computer science teaching, they should work to find ways to improve computer science teacher identity, self-efficacy, and enthusiasm.

CHAPTER III: METHODOLOGY

Chapter III describes the methodology which was implemented to investigate the research questions. A description of the research design and framework will be provided along with details of the study, instrumentation, participants, and sampling. Techniques used in data collection and analysis will be presented as well as an outline of the study procedures. The purpose of the study was to evaluate the impact of tools and curriculum developed through community partnership on high school computer science courses.

As presented in the literature review, the number of computer science graduates exiting the nation's universities annually is far short of the number of workers needed in technology-related positions across the country (Adkins, 2012; Augustine, 2005; Feder, 2012; Fisher, 2015; Grasz, 2013). The impact the computer science worker shortage will have on the nation's economy and national security warrants significant attention be given to this problem. The problem should be researched and addressed on two fronts: increasing interest in computer science among high school graduates and improving retention numbers among college students who have declared a computer science major. This study focused on the former: ways to increase interest in computer science among high school graduates.

High school computer science teachers often lack the resources and curricular insights needed to effectively educate students on technical subjects (Goode, Chapman, & Margolis, 2012; Yadev, et al., 2016). Too often, faculty who primarily focus on business or mathematics are pushed into computer science as instructors due to a lack of qualified computer science teachers (Yadev, et al., 2016). The inexperienced faculty amplify concerns regarding the dearth and deficiency of many computer science faculty development and training programs (Yadev, et al., 2016). Additionally, Exploring Computer Science, or ECS, which is a computer science curriculum developed in Los Angeles and adopted by many school districts nationwide needs to be updated and enhanced (Jed Bodily, personal communication, September 1, 2018). The primary authors of ECS indicated that community partnerships and plentiful, frequently updated faculty training and resources are required to effectively teach the curriculum (Goode, Chapman, & Margolis, 2012). This study is meant to evaluate the impact of additional faculty resources and curriculum developed through community partnerships on high school computer science courses.

3.1 Research Questions

The following research questions guided this study:

1. What are the difficulties faced by Utah K-12 computer science teachers and how can technology tools and community partnerships be leveraged to alleviate some of those difficulties?
2. How are computer science teacher identity, self-efficacy, and enthusiasm of teachers impacted by using a collaborative curriculum portal developed through community partnerships?
3. How are computer science teacher identity, self-efficacy, and enthusiasm of teachers impacted by curriculum enhancements developed through community partnerships?
4. How does knowledge transfer through an interactive curriculum portal impact computer science teacher identity, self-efficacy, and agency in the classroom?

5. What are the perceptions of Utah high school computer science teachers and other key stakeholders regarding the impact of community partnerships on high school computer science courses?

3.2 Building Community Collaboration

The authors of the ECS curriculum cited several lessons learned from their first iterations of the course. Among those lessons learned was an increasingly important reliance on local partnerships (Goode, Chapman, & Margolis, 2012). From improving teacher confidence to faculty recruiting, local partnerships can have a significant impact on teachers and administrators.

Recruiting local schools and teachers to participate in computer science education programs is made easier when professional and material resources needed to launch a successful and academically rigorous computer science class are made available at no initial expense to schools. For administrators, the lure of a university-developed common district curriculum, extensive teacher content and pedagogy support, and class sets of computing resources provides a cohesive framework of support that makes the initial offering of the class less risky (Goode, Chapman, & Margolis, 2012, page 51).

With community partnerships in mind, and to help provide high school computer science teachers with more resources for collaboration and curriculum organization, we organized a collaborative initiative between a local college, a conglomerate of high school teachers and administrators, and representatives of the Utah State Board of Education. The goals of this initiative were to: (a) design and launch an interactive, collaborative web portal to house and organize computer science curriculum for high school computer science teachers, and (b) develop, implement, and assess alternative lesson concepts for the State of Utah's existing Exploring Computer Science course.

The local college in this collaborative initiative was a small, private, computer science-focused institution of higher education located in the Rocky Mountain region of the United States. For the purposes of this study, the college will be known as St. Owensby College. The academic pedagogy at St. Owensby revolves around active learning, student engagement, and problem- and project-based learning. High school teachers and administrators from the state of Utah have encouraged the involvement of St. Owensby in this initiative due to its well-known academic rigor and its effectiveness in creating and delivering engaging computer science curriculum. In recent years, high school teachers and administrators expressed concerns regarding their ability to teach computer science concepts in a way that engages students. Hence, we identified the opportunity to launch an initiative to create a collaborative curriculum portal and develop curriculum enhancements with the goal of better equipping high school computer science teachers to engage students in computer science and improving computer science teachers' self-efficacy and computing identity as a result of that support.

3.3 A Collaborative Curriculum Portal

High school computer science teachers cite a lack of collaborative resources and insufficient direction and training as primary issues impeding their ability to successfully engage students in computer science courses (Yadev, et al., 2016). A collaborative curriculum portal could provide faculty with a way to share ideas and cross train each other as they experience successes and setbacks in the classroom. The portal could also provide easier navigation of curriculum concepts by providing a simple and straightforward user interface. In initial, informal discussions with high school computer science teachers and administrators, it appeared that such a tool would be received very positively.

As part of an initial design phase and informed by the literature, the researcher spent time with prospective users of the system to identify the specific needs related to a curriculum portal. The resulting requirements of the system were as follows:

1. Access
    a. The system should be open and freely available to any user. There will be no fees associated with using the portal or consuming the curriculum content.
    b. The system should require a login to access the curriculum content. The key reasons to require a login are: (a) to obtain email addresses of users so updates to curriculum can be sent out electronically, and (b) to enable persistent user data when posting comments or otherwise collaborating with other users rather than requiring the users to type in their name and demographic data every time they post a comment.
    c. To facilitate login functionality, the system should contain a registration page. Elements of data to collect from the users when registering include:
        i. Username
        ii. Password
        iii. First and last name
        iv. Email address
        v. School
        vi. Primary phone number
        vii. Mailing address

viii. Subjects taught (multi-select from a list of common subjects)

ix. Endorsements earned (multi-select from a list of common endorsements)

x. Opt-in for updates on curriculum, training events, and requests for curriculum/teaching help from other users

2. Curriculum Content

a. Curriculum should be organized in courses. Courses should contain:

i. Title of course

ii. Course description

iii. Link to applicable Utah State Board of Education Strands and Standards webpages

iv. Descriptive text for the Utah Strands and Standards

v. Recommended number of hours for the course

vi. Collaborative partner links to additional resources

b. Courses should contain one or more lesson plans. Lesson plans should contain:

i. Title of lesson plan

ii. Description

iii. Recommended hours

iv. Teacher resources link

v. Student resources link

vi. Link to applicable Utah State Board of Education Strands and Standards webpages

vii. Name of applicable Utah Strands and Standards

viii. List of activities. Each of which should contain:

1. Sequence #
2. Recommended hours
3. Description

ix. List of assessments. Each of which should contain:

1. Sequence #
2. Title
3. Description
4. Teacher notes
5. Rubric
6. Recommended hours
7. Recommended weight

c. The system should be easy to use and allow users to easily navigate courses and lesson plans. Links should be provided to return to previously viewed items via some sort of breadcrumb system.

d. The curriculum content should be directed to teachers and focus on helping them be effective in the classroom by describing how to teach computer science.

3. As users navigate the system, they should be able to find others with whom to collaborate with on courses, specific lesson plans, assessments, and activities. Reasons for users to collaborate could be to post a comment in which they can describe the positives or negatives of a particular lesson, assessment, or activity.

> Users should also be able to "star" a lesson, assessment, or activity and see a total number of "stars" associated with each. Users should also be able to ask questions to the community regarding how to implement particular elements of a course or to seek the help of others in a more general sense. Those who have opted in to answer questions of the community should get said questions forwarded to them electronically, prompting them to login and answer the question if they have the expertise to do so.

The initial launch of the portal was a soft launch with limited curriculum and available only to those teachers who participated in curriculum design and/or portal requirement feedback sessions. This allowed the researcher to solicit feedback from live users in a more controlled environment and to simulate a beta test for the system.

3.4 St. Owensby Computer Science Curriculum Enhancements

The Exploring Computer Science (ECS) curriculum was created in Los Angeles and later adopted by numerous school districts nationwide (Goode, Chapman, & Margolis, 2014). Teachers implementing the curriculum have seen positive results in terms of student engagement and interest in computer science education (Goode, Chapman, & Margolis, 2014). However, Utah's computer science faculty often struggle to make the lessons as engaging as they potentially could be (Jed Bodily, personal communication, September 1, 2018). Utah computer science teachers often lack the technical expertise to teach basic computer science concepts and often struggle to identify ways to capture the attention of students through the provided lesson plans (Jed Bodily, personal communication, September 1, 2018). In looking to enhance the current curriculum for Utah computer science teachers, it is not expedient to attempt to rewrite the program altogether. The existing Exploring Computer Science curriculum is so vast and detailed, and has been so successful nationwide, that a rewrite of the curriculum is not needed. Instead, to provide Utah high school teachers with more options for each lesson plan, alternative activities and projects were created which are meant to be supplementary to the current curriculum.

The Exploring Computer Science curriculum is comprised of six units which focus on human computer interaction, problem solving, web design, programming, computing & data analysis, and robotics (ECS, 2018). In the State of Utah, the introductory course in computer science is known as Exploring Computer Science I and consists of units one (human computer interaction), two (problem solving), and four (programming) from the Exploring Computer Science curriculum (ECS Utah, 2018). The lesson plans with which high school computer science teachers in Utah often struggle are those in units one and two (Jed Bodily, personal communication, September 1, 2018). Gueudet and Trouche (2015) found that computer science teachers typically need significant help in identifying ways to involve students in activities and projects to teach computer science principles. That is also the primary concern among Utah's ECS I teachers (Jed Bodily, personal communication, September 1, 2018). Utah ECS I teachers also struggle understanding the course materials and, just like their peers nationally, they lack professional networks with which to share ideas and from which to learn (Cutts, Robertson, Donaldson, & O'Donnell, 2017; Jed Bodily, personal communication, September 1, 2018; Yadav, Gretter, & Hambrusch, 2015).

The researcher assembled a focus group consisting of Utah teachers who have experience teaching ECS I at the K-12 level, faculty and curriculum developers from St. Owensby College, and representatives of the State Board of Education to provide input into the design of the supplemental lesson plans. The supplemental lesson plans emphasized the use of modern technology and engaging students through projects, group work, and real-life situations that are relative to today's students. Supplemental, alternative lesson plans were created for all lessons in modules one and two of Utah's ECS I course. The supplemental lesson plans adhered to the same strands and standards from the Utah Board of Education as the corresponding lesson plans in the existing curriculum. Additionally, the learning objectives from the existing curriculum were preserved in the supplemental lesson plans. This was imperative given that the intent of the supplemental resources was not to rewrite the curriculum but instead to provide teachers with alternative lesson plans with which to teach the same underlying principles as those found in the existing lesson plans.

3.5 Research Design

3.5.1 Establishing the Paradigm

Mixed methods research refers to a research methodology that involves both quantitative and qualitative data (Wisdom & Creswell, 2013). By using a convergent design technique, the researcher can acquire both quantitative and qualitative data effectively simultaneously to investigate the research questions from various perspectives (Wisdom & Creswell, 2013). The researcher analyzes the data separately and compares trends and themes from both sources (Wisdom & Creswell, 2013). Mixed methods techniques can provide additional depth to findings as well as added validation to research conclusions (Wisdom & Creswell, 2013).

Qualitative research can help craft an understanding of social phenomena and to develop a perspective into why people feel the way they do about aspects of the world around them (Creswell, 2014). This perspective is referred to as a paradigm and represents the world view of the subject (Joubish, Khurram, Ahmed, Fatima, & Haider, 2011). The subject's paradigm can be thought of as the framework for the subject's beliefs and values (Joubish, et al., 2011). It is through this paradigm that the researcher attempts to view the world to better understand the subject's reality (Joubish, et al., 2011).

This study was conducted via mixed methods in which the researcher implemented the constructivist paradigm to gather qualitative data from participants and supplemented, compared, and cross-referenced the qualitative findings with quantitative survey data. When using a constructivist paradigm, a researcher builds a framework to understand the world through human experience (Mackenzie & Knipe, 2006). The subject's feelings, thoughts, and experiences are used to establish the framework so that the researcher is reliant on the subjects' views of each situation (Creswell, 2014). Typically, constructivists will use open-ended questions to seek a clearer, less biased understanding of the subjects' world lens (Creswell, 2014).

The qualitative data collection in this study was grounded in emergent design. In this type of naturalistic inquiry, the researcher builds a hypothesis framework and creates a baseline set of assumptions and, ultimately, questions to ask of participants (Lincoln & Guba, 1985). In each interview with participants, the researcher sought additional insight into emerging themes (Lincoln & Guba, 1985). As interviews were concluded, the researcher compared key points and themes to identify new, common themes which should be identified as future areas of focus (Lincoln & Guba, 1985). The initial framework for this study was grounded in the assumption that Utah computer science teachers require more resources and training and that they want help with ways to improve student engagement in the curriculum for Utah's ECS I course. As themes emerged from interviews with participants, the emergent design allowed the researcher to modify those assumptions and alter the framework for the worldview lens throughout the study.

3.5.2 Type of Study

This study used case study to collect qualitative data through in-person interviews where the researcher investigated participants' worldviews related to the research questions. The case studies in question were Utah-based K-12 classrooms where alternative lesson plans for Utah's ECS I course were implemented. The use of case study allows the researcher to extract common themes from various realities through participant interviews (Joubish, et al., 2011). Merriam (1998) posited that case studies are particularistic, descriptive, and heuristic. Merriam explained further that particularistic describes the notion that case study is meant to focus on specific phenomenon or happenings (Merriam, 1998). Merriam clarified that case study is a descriptive method because the study of events via case study yields a rich and detailed world view of the circumstance being studied (Merriam, 1998). Finally, Merriam indicated that case study is a heuristic technique because researchers can find new meaning and understanding of events being studied by leveraging case study techniques (Merriam, 1998).

This study also used a survey to collect quantitative data to garner further insight into the perspective of participants. Quantitative data is in numerical or statistical format and come as a result of asking closed-ended questions to participants (Creswell, 2014). This form of research is highly structured and survey collection instrumentation are created in advance of the study (Creswell, 2014). Participants in this portion of the study were surveyed on events related to the research questions, specifically, the helpfulness and utility of the curriculum portal and the effectiveness of the curriculum enhancements. Questions for the survey can be found in Appendix B. The survey questions are based on the model outlined by Hung and Yang from Boise State University (2015). Hung and Yang (2015) evaluated the effectiveness of instrumentation in studying the impact of online curriculum on computer science teacher identity and self-efficacy. Upon completion of the survey and the interviews, the researcher attempted to identify common themes and to confirm emergent themes in the quantitative and qualitative data.

3.5.3 Instrumentation

Qualitative data is obtained through conducting interviews with participants, reviewing artifacts, or through observation (Creswell, 2014). According to Creswell, researchers in qualitative studies act as the primary collector of data and do not rely heavily in questionnaires or other instruments (2014). However, a baseline template can be used as a starting point for interviews. The interview template that was used for qualitative research in this study can be found in Appendix A.

Quantitative data is numerical or statistical and is primarily obtained through the use of closed-ended questions to participants (Creswell, 2014). This study used a series of closed-ended questions in a survey that was sent to participants selected as outlined in the following section. The survey can be found in Appendix B.

3.6 Participants and Sampling

Participants in this study were drawn from two distinct populations. For quantitative data, the survey found in Appendix B was emailed to all registered users of the collaborative curriculum portal. Quantitative data comes in numerical and statistical form and including all registered users in the survey may improve the chances of getting statistically relevant data by increasing the number of responders.

Regarding qualitative data, participants were selected through purposive sampling. Purposive sampling is one of the most common ways of collecting qualitative data (Palys, 2008). A type of purposive sampling where the intent is to get a wide spectrum of perspectives is maximum variation sampling (Palys, 2008). When using maximum variation sampling, the researcher attempts to find key metrics within the population such as demographic data and then works to identify participants who cover the widest variety possible in terms of those metrics (Palys, 2008). In this study, the survey found in Appendix B contains a question which asks the responders whether they'd be willing to answer questions in a follow-up interview. The researcher identified a population pool consisting of those responders who answered *yes* to the question regarding interest in follow-up interviews. Among those affirmative responders, the researcher identified the 3-5 responders who had the widest range of answers to the other questions in the survey, including the perceived value of the portal, the perceived value of the curriculum, and other variables such as years of experience and age level taught.

In addition to the maximum variation sampling, the researcher attempted to add an additional 2-4 participants using snowball sampling. In snowball sampling, also known as chain sampling, the researcher asks participants for recommendations regarding other potential participants who may add value to the study (Patton, 1990). In this study, the researcher asked participants who are interviewed if they knew of others who were using the portal and/or the curriculum and who may have helpful additional insight.

3.7 Data Collection

Interviews were conducted in a semi-structured manner. In semi-structured interviews, the researcher starts with open-ended questions and then explores the worldview of the participants through more directed and customized questions as emerging topics are uncovered (Harris & Brown, 2010). To maximize the efficiency of these types of interviews, participants should be in a setting in which they feel comfortable and willing to discuss their feelings freely (Creswell, 2014).

In this study, interviews were conducted in-person at the high school teacher's school or on the telephone, whichever was preferred by the participant. This provided an atmosphere that was at least, in some way, selected by the participant and in which they potentially felt safe, secure, and able to communicate openly. The interviews followed the template found in Appendix A.

3.8 Data Analysis

Throughout the data analysis phase of the study, the researcher followed a linear design as recommended by Creswell (2014). When following this design, the first step is to prepare all data for analysis by transcribing interviews into digital form and consolidating and digitizing all field notes (Creswell, 2014). The next step involves a robust and detailed study of the resulting data in which common themes are identified and extracted (Creswell, 2014). Next is codifying of the data in which the researcher categorizes findings and compared themes, topics, and other ideas for cross-comparison and validation (Creswell, 2014). Next, Creswell recommends creating both broad and detailed descriptions of the major themes that resulted from the codification step. Those descriptions are then used to create a narrative surrounding the central themes in the study (Creswell, 2014). Finally, the researcher interprets the findings to draw conclusions from the study and suggest next steps for future research (Creswell, 2014).

In this study, the qualitative analysis was compared with the numerical and statistical data that resulted from the quantitative survey. Adding quantitative data to the evaluation of qualitative data can help provide added confidence in and generalizability to the findings (Bergman, 2008). In this step, the researcher evaluated the findings from the qualitative study as well as the quantitative statistics and sought to build a relationship between the numerical data and the themes that emerged in the qualitative portion of the study (Bergman, 2008).

After the interviews were transcribed, we removed all identifying information and replaced the interviewee name with a pseudonym in all data files outside a single, password-protected key file. We then reviewed the transcripts in an effort to find any common themes in the data. As themes emerged, we recorded them in a Microsoft Excel document. In addition to themes, sub-themes were also identified and recorded to add an extra layer of granularity to the theme data. Along with each theme and sub-theme, a short description of the comment from the interview was recorded to help the research team identify the context in which the theme arose. A key string from the interview transcript was also recorded with each theme and sub-theme to enable us to easily trace back from the coding Excel file to the transcript itself.

After all theme data was extrapolated from the interview transcripts, we calculated the frequency of each theme and sub-theme. Both theme and sub-theme frequencies were calculated as a total count (the number of times the theme or sub-theme arose in the interviews, including multiple times in a given interview) and an interview count (the number of interviews in which the theme or sub-theme arose, counting only the first time the theme arose in a given interview). Themes and sub-themes which arose and their associated total counts and interview counts are shown below in Table 2.

Table 2. Themes and Sub-Themes Appearing in Interview Transcripts

Theme	Topic	Total Count	Interview Count
Curriculum	Classroom observation	1	1
Curriculum	Discover passion in CS	2	2
Curriculum	Fortnite	9	3
Curriculum	PacMan	28	7
Curriculum	Benefits	1	1
Curriculum	Usage	8	6
Demographic Data	Academic/industry background	4	4
Demographic Data	Age of students	3	3
Demographic Data	Other subjects taught	5	5
Demographic Data	Years teaching CS	4	4
Demographic Data	Years teaching ECS	6	6
Demographic Data	Years teaching k12	3	2
Depth vs. Breadth	CS intro courses	8	4
Difficulties Faced	Can't digest lessons	3	3
Difficulties Faced	Don't understand CS	3	2
Difficulties Faced	Few collaboration opportunities	7	4
Difficulties Faced	Hard to make CS interesting to students	2	2
Difficulties Faced	Other	1	1
Difficulties Faced	Too little time to focus on CS	1	1
Difficulties Faced	Too many resources	2	1
CS Enthusiasm	Affected by curriculum	6	4
CS Enthusiasm	Affected by portal	2	2
CS Enthusiasm	Concerns	5	2
CS Enthusiasm	Of the source	5	3
CS Enthusiasm	Recommendation	0	0
CS Identity	Affected by curriculum	3	3
CS Identity	Affected by portal	1	1
CS Identity	Concerns	10	3
CS Identity	Of the source	9	5
CS Identity	Recommendations	0	0
Impact of Partnerships	General	10	6
Knowledge Transfer	Affected by curriculum	5	4
Knowledge Transfer	Affected by portal	31	7
Knowledge Transfer	Concerns	12	4
Knowledge Transfer	Of the source	0	0
Knowledge Transfer	Recommendations	5	3
Next Steps and Recommendations	General recommendations	6	4
Portal	Benefits	11	6
Portal	Usage	1	1
CS Self-Efficacy	Affected by portal	0	0
CS Self-Efficacy	Affected curriculum	3	2
CS Self-Efficacy	Concerns	16	5
CS Self-Efficacy	Of the source	13	6
CS Self-Efficacy	Recommendations	1	1

Once we identified the frequency of themes and sub-themes, we worked to identify ways to coalesce those themes into overarching topics. These topics then became the core of the findings section discussed later in the paper.

3.9 Procedures

The following serves as a guiding outline of the procedures of the study. The outline illustrates how participants were selected, how interviews were conducted, and how transcriptions were completed.

3.9.1 Participant Selection

Participants in the quantitative data portion of the study completed an electronic survey. The survey was emailed to all registered users of the curriculum portal. The email described the voluntary nature of the survey and reinforced to the recipient that their participation in the study was confidential and anonymous. The email which was to be sent to participants can be found in Appendix C. An information sheet and consent form which disclosed additional information about the survey and participation was attached to the email. That information sheet and consent form can be found in Appendix D.

The survey contained a question which asked the participants if they would be willing to participate further in an interview setting. From the pool of participants who indicated a willingness to be interviewed, the researcher used maximum variation sampling to identify a group of 3-5 participants with whom to conduct interviews. Selection was based on identifying a varied spectrum of demographic data such as years of teaching experience and age of students taught as well as responses to the other survey questions. The email, which was sent to interview candidates, can be found in Appendix E.

3.9.2 The Interview Process

At the beginning of each interview, the information sheet and consent forms were collected. Participants were informed that the interview was to be recorded and transcribed. The researcher reminded the participants of their confidentiality and anonymity in the study and how those measures would be maintained. The researcher also reminded the participants that their participation was completely voluntary and that if at any point during the interview they wished to end their participation, they could do so with no repercussions.

Interviews lasted 30-60 minutes each. The researcher asked questions from the interview template in Appendix A. As emerging themes appeared throughout the interviews, the researcher asked probing follow-up questions to better understand the themes and the participants' perspectives regarding them.

The researcher relied on member-checking throughout the interviews and particularly at the end of the interview. Member-checking is a process in which the researcher repeats what was heard throughout the interview back to the participant and asks whether the participant was understood correctly and provides an opportunity for the participant to correct any inaccuracies (Lincoln & Guba, 1985). The researcher made use of field notes throughout the interviews. These notes were critical in helping the researcher conduct member-checking and record data regarding potential themes.

At the end of each interview, the researcher asked the participant if they knew of others who would add value to the study and who should be interviewed. This type of sampling, called snowball sampling, allows the researcher to add participants who may have beneficial insight into the events being studied.

Once the interviews were completed, they were transcribed into Microsoft Word documents. Themes were identified through content analysis. Conventional content analysis is typically used to describe a phenomenon, such as the reactions of teachers to the curriculum and curriculum portal (Hsieh & Shannon, 2005). The first step in content analysis involves an in-depth reading and understanding of the data (Hsieh & Shannon, 2005). Next, the researcher identifies key words, thoughts, or concepts (Hsieh & Shannon, 2005). The researcher then makes notes regarding first impressions and thoughts during the data analysis (Hsieh & Shannon, 2005). These thoughts evolve into the initial codes and are categorized into groups or clusters (Hsieh & Shannon, 2005). As the researcher continues to identify codes, relationships between categories and subcategories are identified and formulated (Hsieh & Shannon, 2005).

In this study, as themes were identified, they were added to a codebook, kept as an Excel file. As common themes were identified, they were added to the Excel file. Throughout the coding process, the researcher stored brief notes regarding codes identified and correlated thoughts and impressions from the interviews in the codebook. The researcher also quantified codes and themes in the codebook with statistical data regarding how often and how frequently they arose in the interviews. Interviews followed a semi-structured format where an initial set of questions were used. As emergent themes appeared in the study, the researcher asked probing follow-up questions to learn more about the perspective of the participant. After all interviews were complete, the researcher compared the common themes and other data in the codebook to identify trends and central themes for the entire study. The quantitative data was then evaluated and compared against the central themes and findings of the study to help triangulate the findings and improve the validity of the study. The researcher used the quantitative data to identify statistics that either confirmed or contradicted the qualitative findings.

3.10 Context and Limitations

3.10.1 Context of the Study

Throughout the past few decades, there have been numerous calls for STEM education reform from economists, employers, and educators alike (Mara & Rodgers, 2012; U.S. Department of Education, 2013; Watkins & Mazur, 2013). In 2015, there were over 527,000 open computer science jobs and fewer than 60,000 computer science graduates to fill them (National Center for Education Statistics, 2015). The nation is on pace to have 1.4 million open computer science jobs by 2020 and only 400,000 available qualified candidates (Kalil & Jahanian, 2013). This computer science worker shortage represents a significant threat to our nation's economy, standard of living, and our national security (Augustine, 2005; Holdren & Lander, 2012; Launches, 2009; U.S. Department of Education, 2013).

An initiative was launched in the Los Angeles Unified School District to develop a high school computer science curriculum which engages students and piques their interest in computer science (Goode, Chapman, & Margolis, 2012). The resulting program, entitled Exploring Computer Science, or ECS, quickly evolved into a standard at school districts nationwide (Goode, Chapman, & Margolis, 2012). The State of Utah was the first state to adopt the curriculum program statewide (Hu, Heiner, & McCarthy, 2016). The creators of ECS claimed that the program relies on strong community partnerships to be effective (Goode, Chapman, & Margolis, 2012). However, high school computer science teachers report that insufficient training and collaboration resources are impeding their ability to teach computer science concepts (Yadev, Gretter, Hambrusch, & Sands, 2016). In order to improve computer science student engagement and, in turn, computer science interest among students, the teachers need access to improved tools for collaboration and ways to make the curriculum more engaging to students.

3.10.2 Researcher Positionality

When conducting a study, the researcher must endeavor to understand the reality of the participants in the study to be able to frame the opinions expressed by each participant in a context relative to one another. Likewise, it is important that the reader understand the perspective and reality of the researcher to be able to properly assign context to this study.

Academically, the researcher earned a bachelor's degree in computer science from a public state institution, a master's in business administration from a small, private college, a master's in computer science from a public research institution and a doctorate in education from a public research institution.

In his professional career, the researcher worked in the software development field for nearly ten years at both large corporations and small startup companies. Since 2004, the researcher has been working in higher education, specifically, in a private, computer science-focused institution. The researcher has experience teaching computer science at the university level and managing academic departments, programs, and initiatives. The researcher has also spent time working in a university relations capacity in which he would seek to build relationships with employers such as ACS, Bosch, eBay, Fidelity Investments, Nike, and Oracle in order to leverage those relationship into hiring opportunities for graduates and for curriculum collaboration between the institution and the employer.

The combination of experience working for nearly a decade in the software development industry followed by 14 years in higher education provides the researcher with a unique perspective on education and on the computer science worker shortage problem. His experience building collaborative relationships with employers provided the researcher with an insight into the challenges those employers face. An overarching, common theme among virtually all the employers with whom the researcher has interfaced is the continual need to hire more qualified computer science workers. The researcher's perspective and worldview are based on the premise that educators can make an impact on this problem but that it should be driven by collaboration with industry and other educators to provide an industry-driven education that is based on evolving best practices of active learning and student engagement.

3.10.3 Limitations of the Study

This study was conducted in the state of Utah with a small sample of high school computer science teachers. The state of Utah was the first to adopt the ECS curriculum statewide (Goode, Chapman, Margolis, 2012). Utah's school system includes a wide range of rural and urban school districts which pose a challenge to administrators looking to establish a common curriculum program for all districts (Goode, Chapman, Margolis, 2014). The study involved the analysis and evaluation of a collaborative curriculum portal and a series of curriculum alternatives to the ECS program, implemented in the state of Utah.

3.10.4 Assumptions

The major assumption in this study is the premise that if high school computer science courses can be improved, more students will exit high school with an interest in computer science as a college major choice and/or a career. Although this study depends heavily on this assumption, the assumption itself is not unfounded. In fact, administrators involved in the design and launch of ECS in Los Angeles found that the program positively impacted the computer science interests of the students. After taking ECS, the percentage of students who expressed an interest in continuing to study computer science increased from 17% to 43% (Goode, Chapman,

& Margolis, 2012). The administrators also found that approximately 60% of the students reported an interest in working in the computer science field (Goode, Chapman, & Margolis, 2012).

CHAPTER IV: ST. OWENSBY CURRICULUM ENHANCEMENTS

Part of this study revolves around the analysis of the impact of curriculum enhancements that were created through a partnership between administrators and faculty at St. Owensby College, Utah K-12 computer science teachers, and the Utah State Board of Education. This section presents details around that curriculum and its design.

4.1 Context and Rationale

In the spring of 2018, we created a partnership with faculty and administration at St. Owensby College, members of the Utah State Board of Education and local high school computer science teachers with the goal of identifying ways to improve K-12 computer science education in Utah (Jed Bodily, personal communication, April 18, 2018). After discussions with various K-12 computer science teachers and members of the board of education as well as those participating in the partnership, the first project was identified: creating curriculum enhancements for teachers to use as alternative lesson plans in Utah's ECS I course. ECS I was chosen because of the consistent concerns from those teaching the subject regarding the existing curriculum (Jed Bodily, personal communication, April 18, 2018; Brandon Jacobson, personal communication, November 19, 2018). Many of Utah's ECS I teachers felt the existing curriculum did not keep student interest, focused on outdated technology or concepts, and lacked principles of active learning or project-based learning (Jed Bodily, personal communication, April 18, 2018; Brandon Jacobson, personal communication, November 19, 2018).

Utah's ECS I course is generally a student's first foray into the world of computer science education (Brandon Jacobson, personal communication, November 19, 2018). As such, the partnership agreed that the course needed to be as engaging and interesting as possible. We determined that the goal of any curriculum enhancements for ECS I should be designed to capture the interest of the students and build a passion for learning more about computer science. The course curriculum currently adopts modules 1, 2, and 4 from the nationwide ECS curriculum developed in Los Angeles, California. The focus of module 1 is human computer interaction and in that module, students learn about computers, how they work, how they process information, and ways in which humans interact with them. The focus of module 2 is problem solving and there the students learn about problems, the problem-solving process, how to break down large problems and solve them, binary number systems, and various algorithms and data structures. The focus of module 4 is an introduction to programming and in that module, students learn introductory programming skills using the Scratch programming language. Generally, Utah's K-12 computer science teachers find module 4 to be working well and capturing student interest because students are getting hands on experience writing code (Jed Bodily, personal communication, April 18, 2018; Brandon Jacobson, personal communication, November 19, 2018). Hence, we focused on developing alternative or supplementary lessons for modules 1 and 2.

Throughout the discussions with members of the partnership and others in the Utah K-12 computer science education community, we felt the problems with the existing curriculum went beyond delivery methods of the course content and into the learning objectives themselves. For example, each of the 11 K-12 teachers involved in the partnership unanimously pointed to the first half of unit 2 as a major pain point. The first four lessons of that module, which encompass nine days of the class, revolve around the problem-solving process. Each of the teachers felt that they were able to effectively teach the problem-solving portion of the course in half that time or less, which left the teachers with an extra week of class to fill with some other content. Additionally, teachers felt the problem-solving course content was especially boring to their students and in need of an overhaul. Many members of the partnership, particularly those involved in the classroom at the K-12 level, expressed an opinion that the course outcomes and objectives needed a refresh as well, and they recommended that some current outcomes be removed and other new ones be added. With such complex design problems in mind, we had two choices: make a completely new curriculum from scratch by applying sweeping changes or stick to the rough outline of the current curriculum and build per-lesson alternative teaching ideas. Ultimately, we went with the latter option and created a set of alternative lesson plans for each lesson in ECS modules 1 and 2. Although we felt the curriculum may need an overhaul of larger scope, we felt that in order to gain traction and support from teachers, administrators, and the board of education, helping first in a smaller, less disruptive way would be more prudent. Another factor leading to that decision was statewide testing on ECS I subjects. If we were to recommend larger-scale changes to curriculum and those changes were accepted by the board of education, the statewide tests would also require an overhaul to match the new curriculum. The need to overhaul Utah’s ECS I curriculum on a larger scope may still exist but it was outside the scope of our initial curriculum project in the St. Owensby partnership and it is outside the scope of this study.

Another decision we made in the interest of encouraging adoption of the curriculum changes was to integrate with the existing ECS I curriculum as much as possible. We felt that teachers and administrators would be more likely at encourage adoption of our curriculum enhancements if they were presented as just another way to teach the same lesson, yielding the same outcomes. Hence, the curriculum enhancements designed by the St. Owensby partnership are not meant to replace any of the existing ECS I curriculum resources or lesson plans. Instead, they are meant to go hand-in-hand with the existing curriculum with the idea that a teacher could look at a given module and lesson in the existing curriculum, see how it is to be taught, and then look at the same lesson in the St. Owensby curriculum and see another way the same lesson can be taught. These design decisions were instrumental in gaining immediate and full support from members of Utah's Board of Education (Brandon Jacobson, personal communication, November 19, 2018).

4.2 Design Methodology

Modules 1 and 2 of Utah's ECS I curriculum draws from modules 1 and 2 of the nationwide ECS curriculum. Those modules are comprised of 17 lessons in total (eight in module 1 and nine in module 2). The collaborative partnership met four times throughout the summer of 2018 with the goal of recommending curriculum enhancements for each of the 17 lessons. Dates of the meetings were: June 4, June 26, July 12, and August 13. Before each of the meetings, Jed Bodily, Manager of High School Relationships at St. Owensby College worked with Brandon Jacobson of the Utah State Board of Education to publicize the next meeting to all of Utah's K-12 computer science teachers. Attendance at each meeting included members of St. Owensby administration, St. Owensby faculty, various Utah K-12 computer science teachers, and members of the Utah State Board of Education. Meetings took place at the St. Owensby College campus.

The meetings focused on presenting a layout of existing ECS lesson plans and fostering collaborative discussion regarding improvements. Generally, the dissection of a given lesson plan at the meetings was conducted according to the following format:

1. A member of the St. Owensby administration distributed copies of the ECS lesson plan and read through the lesson.
2. Members of the partnership, particularly those teaching ECS I at the K-12 level, were asked to discuss what they liked about the lesson, what students seem to like, and what about the lesson was currently working well.
3. Members of the partnership, particularly those teaching ECS I at the K-12 level, were asked to discuss what they did not like about the lesson, what students seemed to dislike, and what about the lesson was currently not working well.
4. A brainstorming session was then conducted in which high-level ideas for improving the lesson were written on a whiteboard. Ideas were meant to revolve around concepts of active learning and project-based learning while maintaining consistency with the lesson's learning objectives.
5. After the brainstorm session, one or two high-level ideas were identified as candidates for the new lesson and a discussion was held to create a rough outline for delivery of that lesson.
6. After the rough outline was created, the partnership moved on to the next ECS lesson.

After each meeting, we expounded on the rough outline for each lesson, elaborating on the details of the lesson and taking caution to elucidate important technology-related aspects of the lesson to help teachers more easily understand the lesson concepts and more effectively teach those concepts. The result was a lesson plan in the St. Owensby curriculum enhancement format.

4.3 Lesson Plan Format

After some debate in the first partnership meeting, we adopted a standard format for the St. Owensby curriculum enhancement lessons. Although more radically different formats were discussed, ultimately the partnership agreed that adopting a format with attributes similar to the existing ECS curriculum would be preferred. Such a format, we believed, would help encourage adoption because the format of the lesson plan would already be familiar to teachers and it would enable teachers to quickly look at the ECS lesson and the St. Owensby lesson and identify similarities and differences. The rest of this section describes the St. Owensby lesson plan format. All St. Owensby lesson plans can be found in Appendix F.

To minimized deviation from the ECS lessons, the title of each St. Owensby lesson plan consisted of the module number and the lesson plan days from the ECS curriculum. For example, a sample title from the first lesson in the course would be: Module 1 Days 1-2. We felt that using this format for the title would minimize confusion for teachers as they looked for alternative ideas for teaching a given lesson in the ECS curriculum. Under the title, at the top of the lesson plan, a summary of the number of hours required to teach the lesson was given.

The ECS lessons each have a topic description, which describes the lesson theme and provides a brief overview. ECS lessons also have a list of objectives for that lesson, which describe what the student should be able to do after completing the lesson. As our goal was to create supplemental resources and alternative lesson plans to teach the same curriculum rather than to create radically changed curriculum, we decided to keep the same topic description and lesson objectives in our curriculum enhancements. However, due to copyright concerns, we excluded those from our lesson plan format. Instead, at the top of teach St. Owensby lesson plan, we referred the reader to the appropriate pages in the ECS curriculum to get the topic description, objectives, and to see the ECS outline for the lesson. A sample reference for instructional days 1-2 is shown below:

> *Instructional Days: 1-2 (Topic description, objectives, and ECS outline can be found on pages 30-31 of the ECS 8 curriculum)*

Following the reference to the ECS topic description and objectives, a disclaimer is found in each lesson plan. The disclaimer is meant to clarify the rationale for the St. Owensby lessons and the fact that the lessons are not meant to replace ECS curriculum or deviate from that curriculum but instead meant to supplement and provide alternative ideas for teaching the same concepts. The disclaimer for each lesson is the same and reads as follows:

> *Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.*
>
> *This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.*

The remainder of the lesson plan breaks down the lesson into an outline, a list of student activities, and teaching/learning strategies. Again, to minimize confusion and facilitate adoption, these sections are also consistent with the ECS lesson plan format. The outline lists the various components of the lesson with a summary of anticipated time to deliver each component. The student activities list illustrates the types of activities to which students will be exposed during the lesson. These may include group presentations, individual work, etc. Finally, the teaching/learning strategies section breaks down the lesson components from the outline in detail, explaining to the teacher how to deliver the lesson. Based on feedback from the K-12 teachers in the partnership, we made the decision to put all information required to teach the lesson in the teaching/learning strategies section, including links to external resources. Some of the teachers in the partnership expressed concern with the ECS format which often requires teachers to jump back and forth in a large PDF document to find all instructions and resources for a given lesson. We felt by putting all the details in one section we could alleviate some of those concerns. The St. Owensby lesson plan format can be examined along with all lessons from the St. Owensby curriculum in Appendix F.

4.4 Lesson Plan Summary

Of the 17 lessons found in modules 1 and 2 of the ECS I curriculum, 12 of them were redesigned and presented as alternative lesson plans. The remaining five lesson plans were all deemed to be working sufficiently well by the K-12 members of the partnership and we decided to focus on improving the other 12 lessons rather than recommending changes to lessons that appear to not need revision. The five lessons which for we did not create alternate plans were all found in module 2 and they are: Day 3, Days 13-14, Days 15-16, Day 17, and Days 18-21. To maintain consistency in the St. Owensby lesson plan listings and to minimize confusion, we decided to create a lesson plan for each of those days that did not need revision. However, those lesson plans contain the following elements: a lesson plan title, the number of hours to deliver the lesson, the disclaimer found in all St. Owensby lessons, and the following line of text: *In our* *design meetings with Utah ECS I teachers, the consensus from the teaching community was that this lesson in the ECS I curriculum works great and did not need additional resources.*

We felt that by creating a lesson plan for those days, even though the lesson plan provides no additional content, would be helpful to teachers who were looking for alternatives for a particular lesson. We also felt that it was preferred to list all ECS module 1 and 2 lessons in the St. Owensby curriculum rather than having five lessons which simply did not exist in the St. Owensby content.

4.5 Summary

This chapter described the context and rationale for creating the St. Owensby curriculum enhancements. The design methodology, including a description of the format of the St. Owensby partnership meetings, was presented. An outline of the lesson plan format which is standard across all St. Owensby lessons was also discussed. The full content of the St. Owensby curriculum enhancements can be found in Appendix F. The next chapter will present the curriculum portal which was developed to store the curriculum enhancements online.

CHAPTER V: CURRICULUM PORTAL

Part of this study revolves around the analysis of the impact of a curriculum portal built as a framework to store the St. Owensby curriculum enhancements. The portal was designed as part of a partnership between administrators and faculty at St. Owensby College, Utah K-12 computer science teachers, and the Utah State Board of Education. This section presents details around the curriculum portal.

5.1 Context and Rationale

One of the key complaints we had heard from members of the partnership who taught ECS I was the cumbersome navigation of curriculum resources. Version 8 of the nationwide ECS curriculum, which can be downloaded in various formats which are used for printing, filing, or larger vs. condensed images, ranges from 148 to 295 pages in a single .PDF file. Teachers find that resource, as well as many others available to them online, difficult to navigate, which increases the challenge to comprehend those resources (Jed Bodily, personal communication, April 18, 2018; Brandon Jacobson, personal communication, November 19, 2018). As a result, we wanted to make the St. Owensby curriculum enhancements in some way more navigable, digestible, and accessible. After a discussion among the partnership, we concluded that the curriculum should be available online on a curriculum portal website which would enable the user to access individual lesson plans and associated resources quickly and easily.

5.2 Requirements and Constraints

After some discussion within the partnership, particularly with those teaching K-12 computer science, an initial set of requirements was developed. Those requirements were:

1. Access
    a. The system should be open and freely available to any user. There will be no fees associated with using the portal or consuming the curriculum content.
    b. The system should require a login to access the curriculum content. The key reasons to require a login are: (a) to obtain email addresses of users so updates to curriculum can be sent out electronically, and (b) to enable persistent user data when posting comments or otherwise collaborating with other users rather than requiring the users to type in their name and demographic data every time they post a comment.
    c. To facilitate login functionality, the system should contain a registration page. Elements of data to collect from the users when registering include:
        i. Username
        ii. Password
        iii. First and last name
        iv. Email address
        v. School
        vi. Primary phone number
        vii. Mailing address
        viii. Subjects taught (multi-select from a list of common subjects)

ix. Endorsements earned (multi-select from a list of common endorsements)

x. Opt-in for updates on curriculum, training events, and requests for curriculum/teaching help from other users

2. Curriculum Content

   a. Curriculum should be organized in courses. Courses should contain:

      i. Title of course

      ii. Course description

      iii. Link to applicable Utah State Board of Education Strands and Standards webpages

      iv. Descriptive text for the Utah Strands and Standards

      v. Recommended number of hours for the course

      vi. Collaborative partner links to additional resources

   b. Courses should contain one or more lesson plans. Lesson plans should contain:

      i. Title of lesson plan

      ii. Description

      iii. Recommended hours

      iv. Teacher resources link

      v. Student resources link

      vi. Link to applicable Utah State Board of Education Strands and Standards webpages

      vii. Name of applicable Utah Strands and Standards

viii. List of activities. Each of which should contain:

1. Sequence #
2. Recommended hours
3. Description

ix. List of assessments. Each of which should contain:

1. Sequence #
2. Title
3. Description
4. Teacher notes
5. Rubric
6. Recommended hours
7. Recommended weight

c. The system should be easy to use and allow users to easily navigate courses and lesson plans. Links should be provided to return to previously viewed items via some sort of breadcrumb system.

d. The curriculum content should be directed to teachers and focus on helping them be effective in the classroom by describing how to teach computer science.

3. As users navigate the system, they should be able to find others with whom to collaborate with on courses, specific lesson plans, assessments, and activities. Reasons for users to collaborate could be to post a comment in which they can describe the positives or negatives of a particular lesson, assessment, or activity. Users should also be able to “star” a lesson, assessment, or activity and see a total

number of “stars” associated with each. Users should also be able to ask questions to the community regarding how to implement particular elements of a course or to seek the help of others in a more general sense. Those who have opted in to answer questions of the community should get said questions forwarded to them electronically, prompting them to login and answer the question if they have the expertise to do so.

5.3 Development Methodology

Given the interest in the curriculum and the associated portal from potential users (K-12 computer science teachers) we decided to use an agile development methodology in the creation of the portal. Although elements of agile software development had been in place for years, the concept was formalized in 2001when a group of 17 technology leaders met at a Utah ski resort to discuss ways to improve the process of software development (Beck, Beedle, Van Bennekum, Cockburn, Cunningham, Fowler, & Kern, 2001). Agile software development represents a change from previously existing and popular software methodologies in four key areas: Agile software development prioritizes:

- Conversation and interaction between all parties over processes and tools
- Prototypes and working software over comprehensive documentation
- Collaboration with customers over contract negotiation
- Responding to change and adapting over blindly following a plan (Beck, et al., 2001).

In order for agile software development to work properly, continual interaction with users is needed (Beck, et al., 2001). The K-12 computer science teachers involved in the St. Owenbsy partnership agreed to be available for continual feedback throughout the development process. We held an initial planning meeting in March of 2018 to determine a set of system requirements (listed in the previous section). At that meeting and in the spirit of agile software development, the decision was made to not create more in-depth requirements on paper and instead to build a working prototype as fast as possible and review it with users to determine next steps.

5.4 Initial System Design

The original plan for the system was to use a model-view-controller (MVC) architecture. In MVC, key components of a web application can be isolated and compartmentalized, minimizing the impact on other components when one aspect of the system changes (Khan, 2016). The model layer is accountable for maintaining the system data. The view is used to display that data to the user. The controller is used to control the interaction between the view and the model layers. A typical AngularJS MVC architecture can be seen in Figure 5.

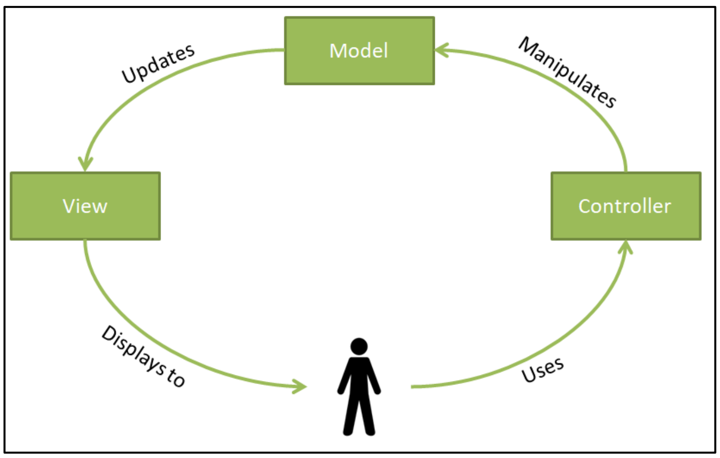


Figure 5 –AngularJS Common MVC Architecture

We initially decided to place information regarding courses, lessons, and resources in JavaScript Object Noation (JSON) files. A database would be used to store login information as well as comments created on the website. A file server would be used to store JSON files. Each of these would be read from and written to by a data model layer, which would send data for the user to see through a view layer.

The development plan was to build front-end web pages in AngularJS 6.1 and Node.js. Angular is a popular web development framework for several reasons. First, it facilitates quick two-way data binding in which the view and the model instantly share changes with one another without developer intervention (Rajput, 2016). Second, AngularJS allows developers to create prototypes rapidly with significantly less code than in other frameworks (Rajput, 2016). Third, AngularJS allows developers to assign special behaviors to the Document Object Model (DOM) through directives, allowing code for specific functions to be reused repeatedly and providing better code organization. Additionally, AngularJS is extremely popular and has a very strong community providing support and tools to help other developers. Another factor in determining to use AngularJS for the web portal was the current expertise we had in developing in that framework.

We decided to use SQL Server as the database because of its robust SQL capabilities and its management tools. Structured Query Language (SQL) is also a language that, according to the K-12 computer science teachers in the partnership, would be more likely used among the teaching community. Hence, if we wanted to facilitate teachers' technical contributions to the project in the future, SQL Server was a preferred choice.

We decided to use ASP.NET as the controller layer for similar reasons. Teachers expressed that would be a technology with which some of the more technical teachers in the community would be familiar. It was also a tool with which we were familiar. Additionally, ASP.NET also helps developers build faster and more efficiently than many other frameworks (Wiseley, 2018).

An architectural diagram of the original plan for the system can be seen in Figure 6:

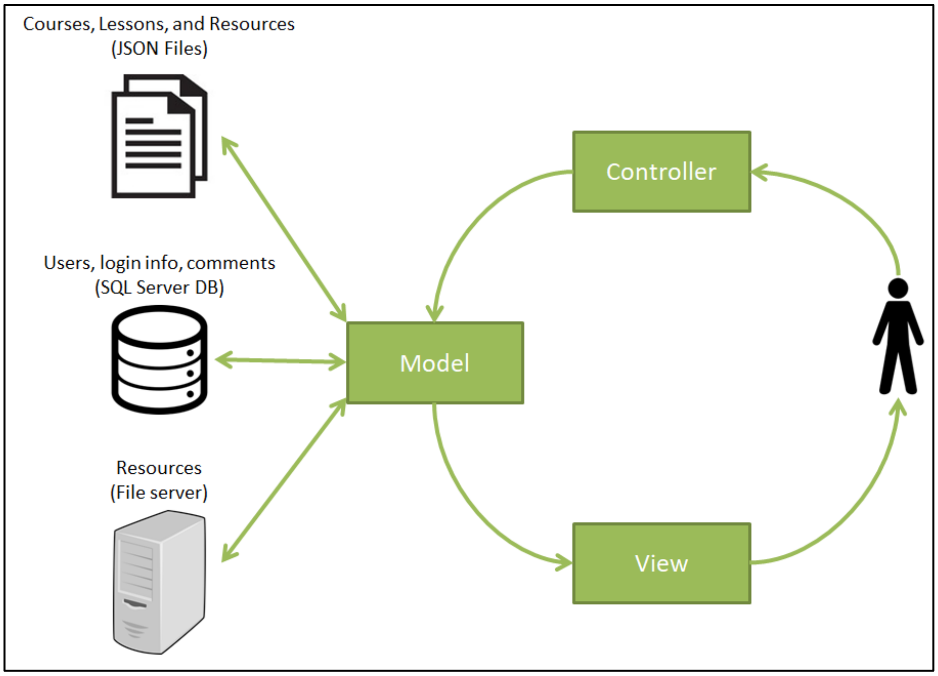


Figure 6 – Initial Portal System Architecture

The database design features five tables. The Course table contains the details of a course including the tile, description, and information about the corresponding state strands and standards. Course resources are stored in the CourseResource table which links to the Course table through the CourseID. The Lesson table contains the title of the lesson and details of the lesson plan for that lesson. Comments are stored in the Comment table and are linked to the lesson through the LessonID. Finally, the UserProfile table contains the user information such as name, email, etc. A database diagram can be found in Figure 7.

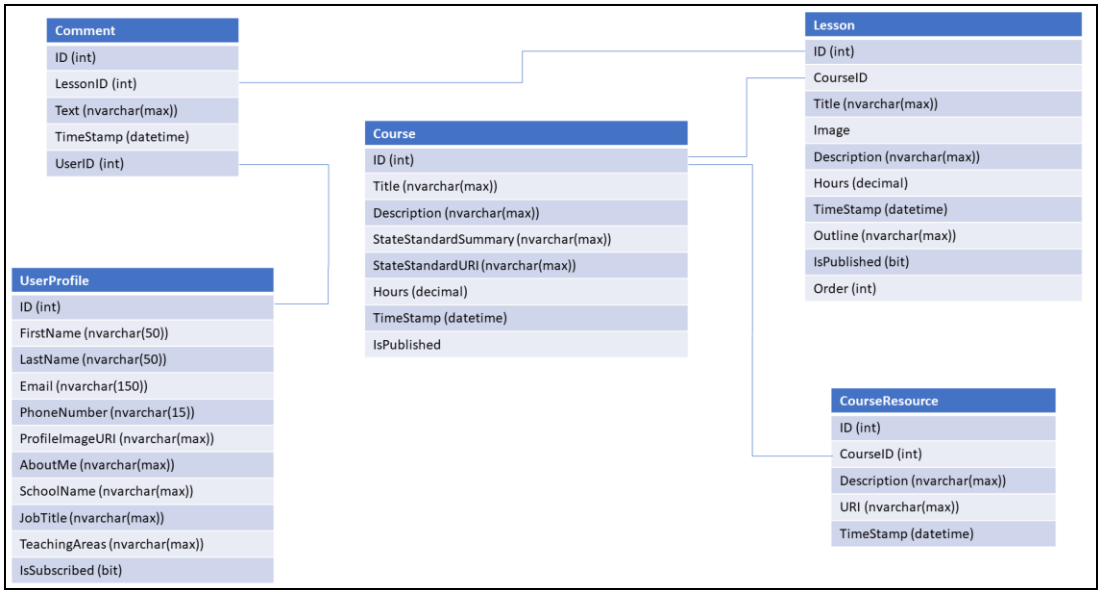


Figure 7 – Portal Database Diagram

5.5 Revised System Design

In May 2018, we finished the initial prototype for the portal. In a subsequent design and review session, the prototype was live for users to experience. We recorded user feedback by watching users interact with the system and then by interviewing them about their experience with the system afterward. Our goal was to identify areas where the users felt comfortable with the system and areas that needed improvement either in functionality, look and feel, intuitiveness, or user preferences.

The biggest shift in system design was made during this meeting. The users found creating JSON files for their lesson plans extremely difficult. Because technical users had a difficult time editing the JSON files, we felt that it would be nearly impossible for non-technical users to do so. We ended up abandoning the JSON file concept to store course and lesson plan data. Instead, we shifted to allow users to create lesson plans in Word documents, something with which we felt all users would be very familiar. We then decided to convert the Word documents to Markdown, a lightweight markup language which easily supports HTML. By downloading a Microsoft Word plugin called Writage, Word files can easily and instantly be converted to markup. We decided to create an admin section on the portal through which administrators could upload markup files for courses or lessons as well as upload course resources such as images and other files. The resulting change removed the JSON component in the data layer and the system architecture diagram was modified as shown in Figure 8:

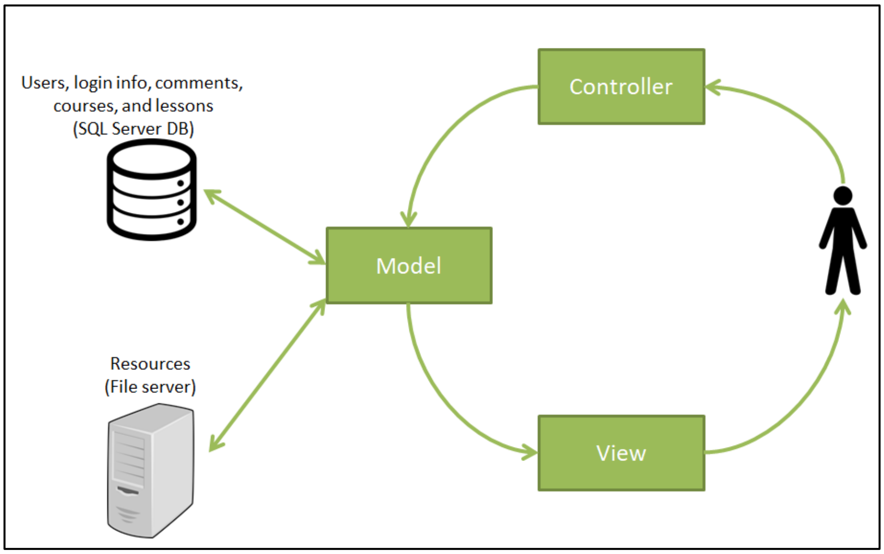


Figure 8 – Revised System Architecture

5.6 User Interface Revisions

The user interface went through two major revisions, one in the May review meeting and another in a review meeting in July 2018. In May, users identified a host of user flow concerns related to a lack of a standard user interface (UI) template which made navigation from page-to-page cumbersome and difficult. Without a standard look to all the pages, users were guessing where to click to move to the next page or to move back to the previous. We were also planning on using a variety of course resources including activities, assessments, teacher resources, and student resources. In the May meeting, users informed us that those were causing confusion and when they were looking for resources they felt it would be preferable to have all resources in the same location, regardless of type. That change affected the database because we had previously used multiple tables for resources and now we were going to consolidate those to one table, as has been shown in the database diagrams in previous sections. The change also significantly streamlined and simplified the UI because only one listing of resources would be needed per course.

In July, a much more refined prototype was shown to users. Again, notes were recorded by watching users interact with the system and by interviewing them afterward. The biggest takeaway from the July session was that users found the system to be functional and helpful, but that it required too many clicks to navigate the course. There was a page for each course and then within a course there was a page for each lesson. Resources were on another page and comments on yet another. There were no breadcrumb links to facilitate users moving back to previous pages on the site. The UI was again greatly simplified in this meeting to have a single page for each course which would consist of a list of lesson plans and a list of resources for that course. Each lesson plan would have a page with the comments on the bottom of the lesson plan page. Users felt that this design would represent a significant improvement and would give teachers the flexibility they need to navigate the system easily and quickly.

5.7 Final User Interface

Several other design review meetings were held throughout the summer of 2018. This section presents the final user interface for the portal website.

When attempting to login to the system, users will be presented with the page shown in Figure 9. Users will then either register for the system by entering their email and password or login using existing credentials.

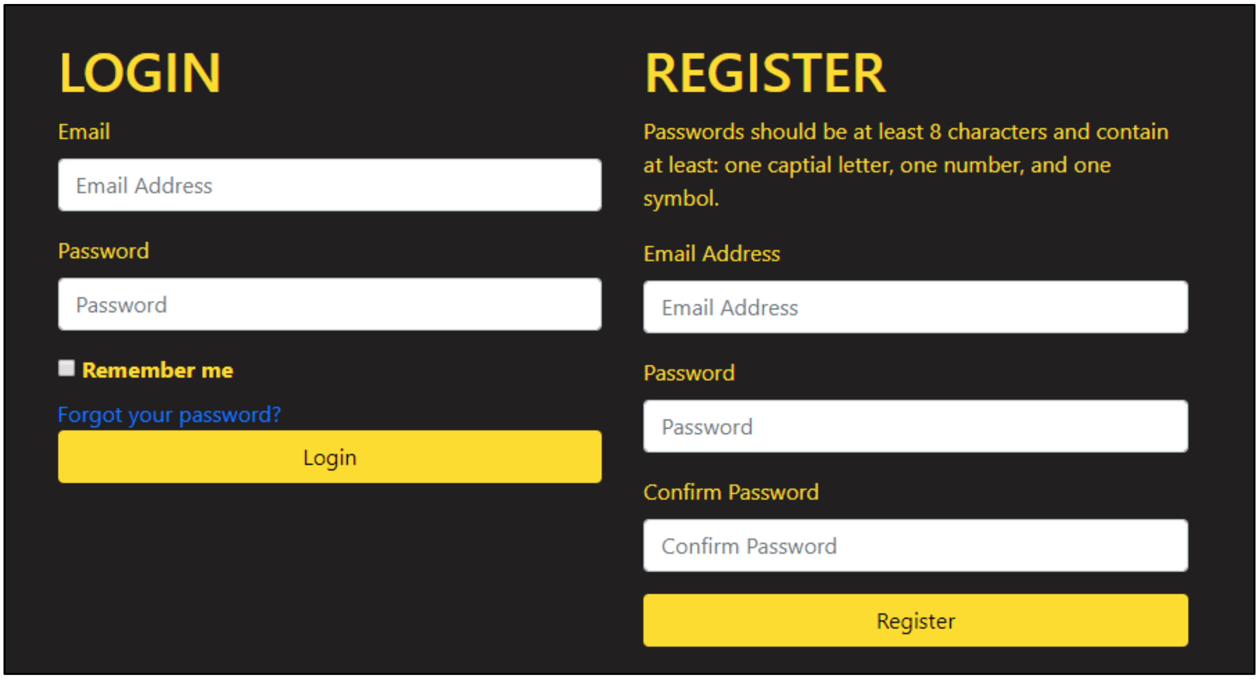


Figure 9 – Portal Login/Registration Screen

Once logged in, users will see the courses page in the portal. This portal shows all current courses found in the portal and can be seen in Figure 10

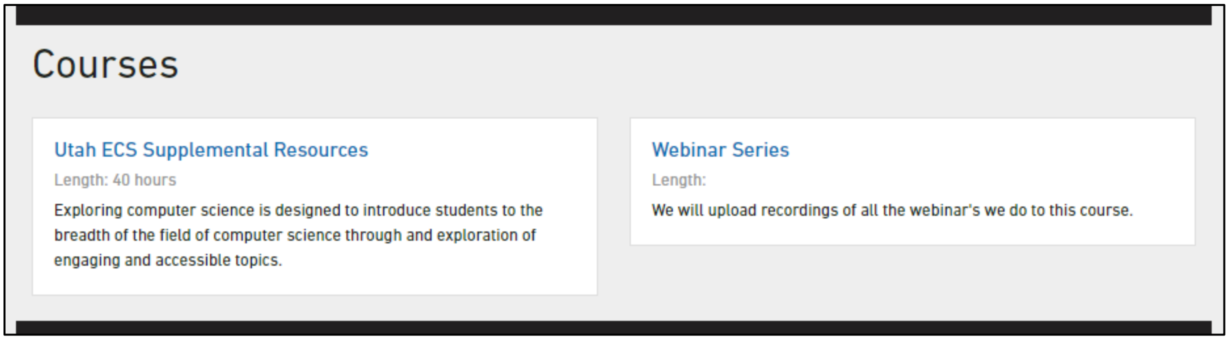


Figure 10 – Courses Page of the Portal

When clicking on a course in the courses page, users will be taken to the page for that course. The course page will show a listing of the lessons for that course with a summary of the time required for that lesson and a brief description of the lesson. The list of all resources for the course will be shown as well as a link to the corresponding strands and standards from the State of Utah. The course page can be seen in Figure 11.

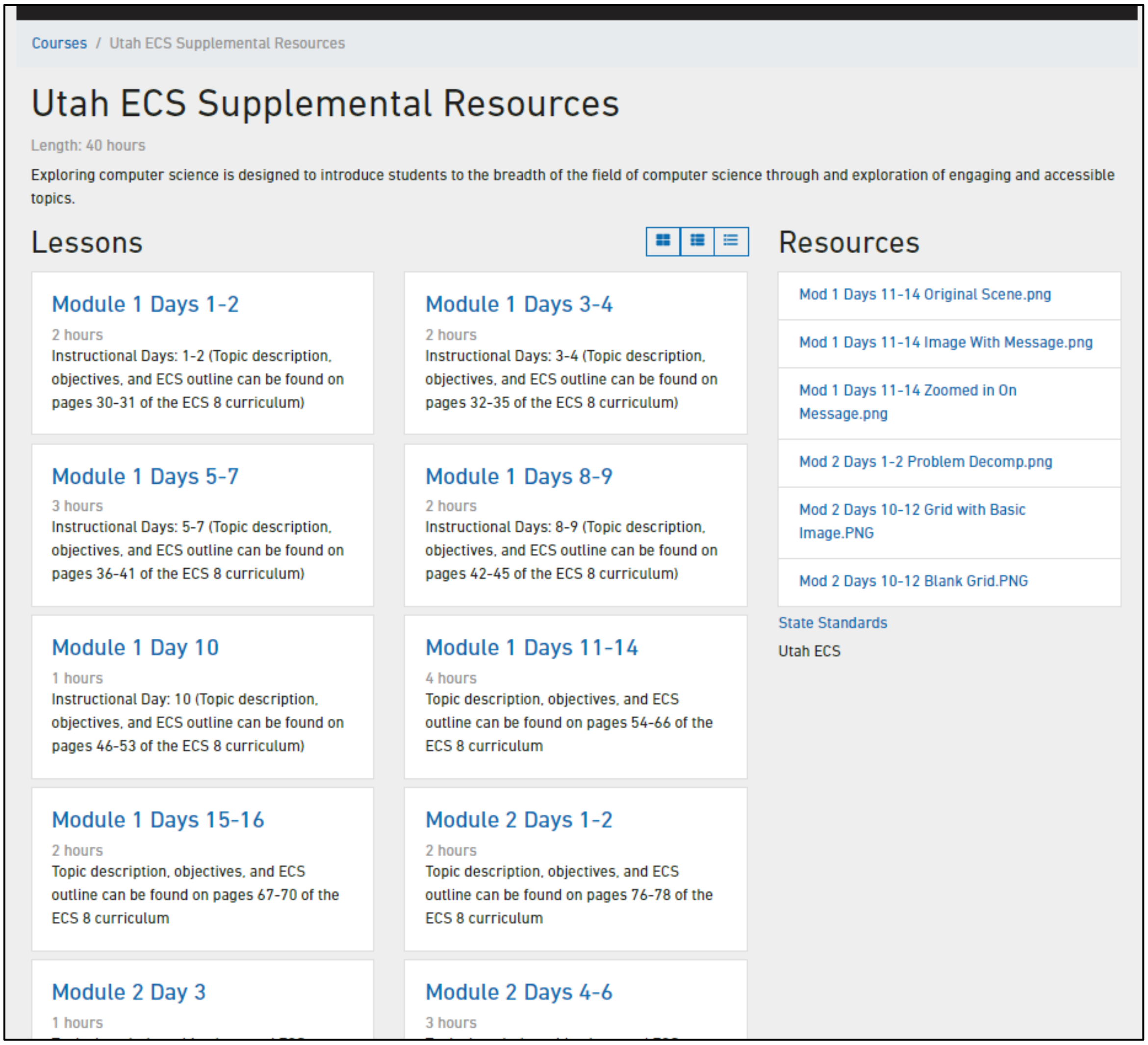
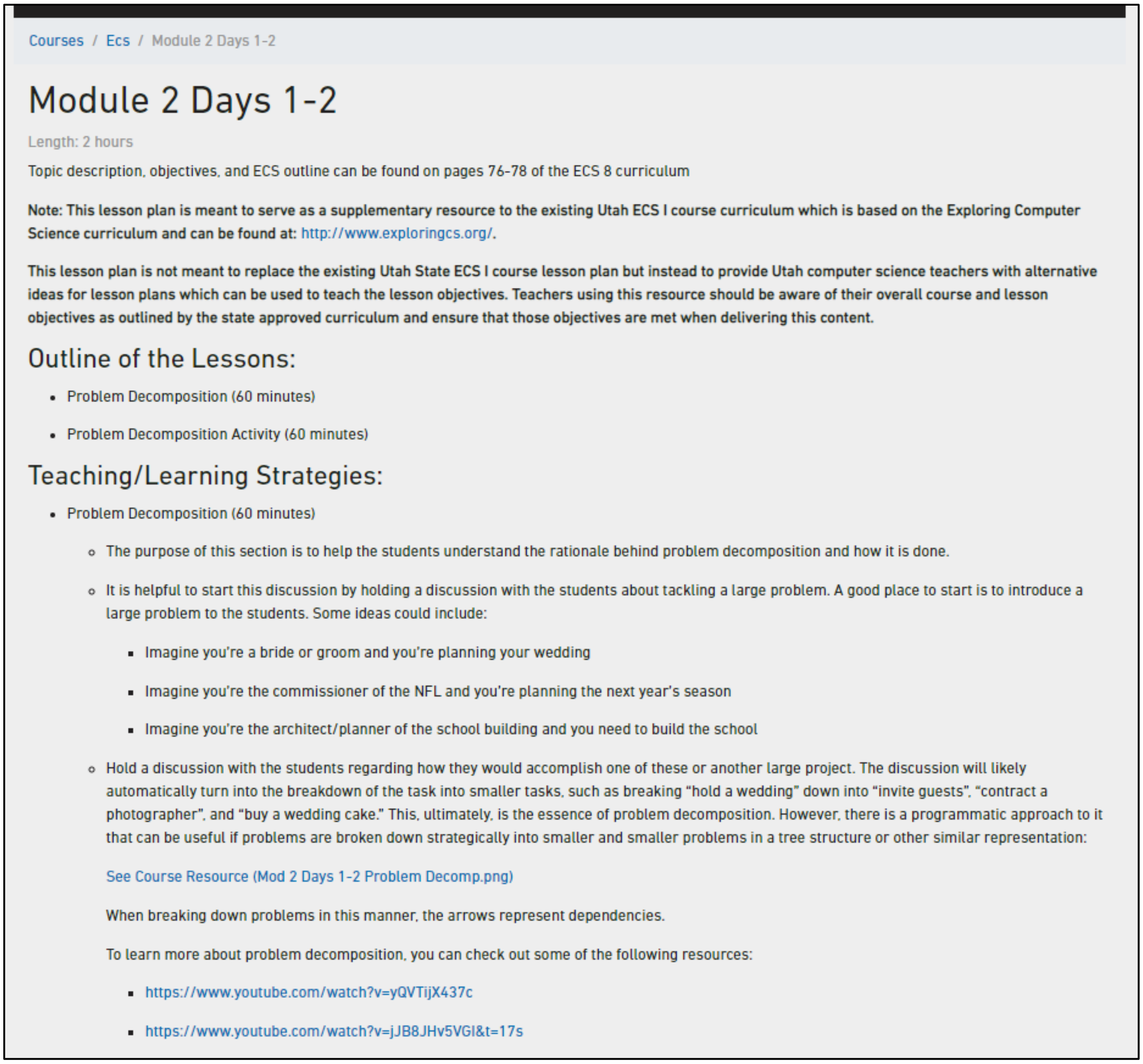


Figure 11 – Course Page of the Portal

Clicking on a resource in the course page will download that resource to the user's computer. Clicking on the State Standards link will take you to the Utah State government website describing the strands and standards for that course. The breadcrumbs at the top of the pages throughout the site will return you to the page on which the user clicked. When clicking on a lesson in the courses page, the user will be presented with the lesson page which displays all content for the selected lesson. The lesson page can be seen in Figure 12.



Courses / Ecs / Module 2 Days 1-2

Module 2 Days 1-2

Length: 2 hours

Topic description, objectives, and ECS outline can be found on pages 76-78 of the ECS 8 curriculum

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

Outline of the Lessons:

- Problem Decomposition (60 minutes)
- Problem Decomposition Activity (60 minutes)

Teaching/Learning Strategies:

- Problem Decomposition (60 minutes)
    - The purpose of this section is to help the students understand the rationale behind problem decomposition and how it is done.
    - It is helpful to start this discussion by holding a discussion with the students about tackling a large problem. A good place to start is to introduce a large problem to the students. Some ideas could include:
        - Imagine you're a bride or groom and you're planning your wedding
        - Imagine you're the commissioner of the NFL and you're planning the next year's season
        - Imagine you're the architect/planner of the school building and you need to build the school
    - Hold a discussion with the students regarding how they would accomplish one of these or another large project. The discussion will likely automatically turn into the breakdown of the task into smaller tasks, such as breaking "hold a wedding" down into "invite guests", "contract a photographer", and "buy a wedding cake." This, ultimately, is the essence of problem decomposition. However, there is a programmatic approach to it that can be useful if problems are broken down strategically into smaller and smaller problems in a tree structure or other similar representation:

      See Course Resource (Mod 2 Days 1-2 Problem Decomp.png)

      When breaking down problems in this manner, the arrows represent dependencies.

      To learn more about problem decomposition, you can check out some of the following resources:

        - https://www.youtube.com/watch?v=yQVTijX437c
        - https://www.youtube.com/watch?v=jJB8JHv5VGI&t=17s

Figure 12 – Lesson Page of the Portal

At the bottom of each lesson, the comments portal is shown which allows users to add their expertise and experiences to the portal to facilitate knowledge transfer and collaboration among teachers. The comments section can be seen in Figure 13.

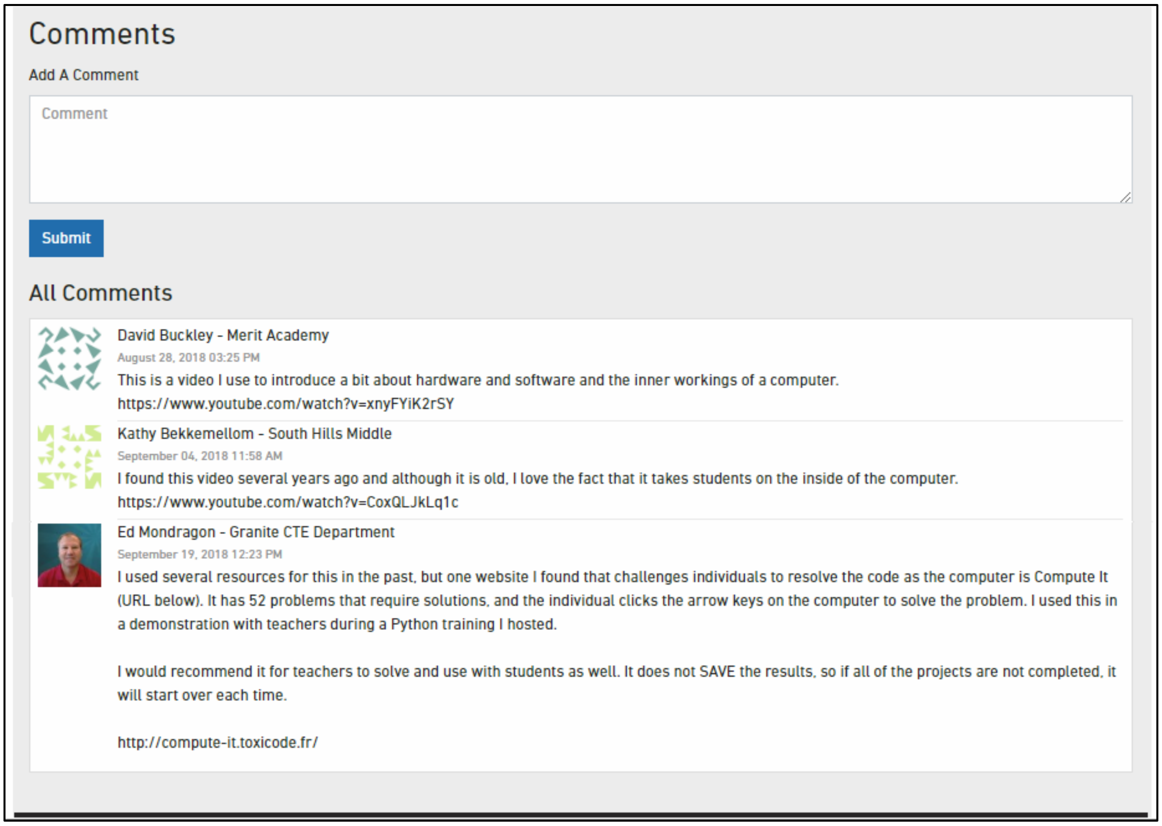


Figure 13 – Comments Section of the Lesson Page of the Portal

An admin section of the website exists where administrators can upload or create new courses, lessons, and resources. The admin portal can be seen in Figure 14

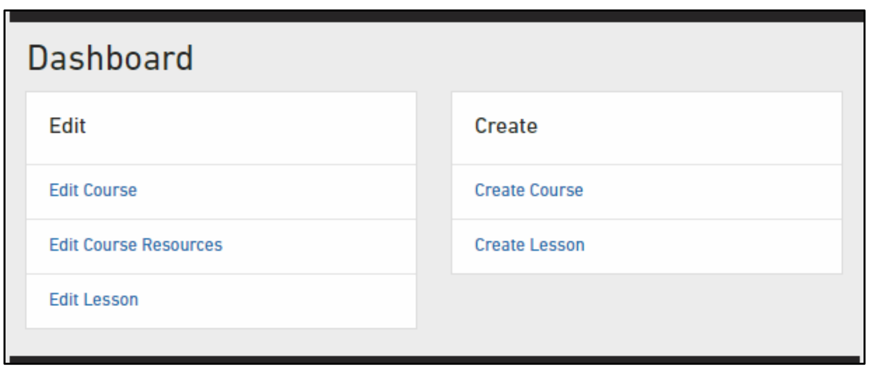


Figure 14 – Admin Section of the Portal

Each of the sections of the admin section function in a similar manner. When editing, the user selects the course, lesson, or resource to edit and is presented with textboxes to modify the information for that course, lesson, or resource. When creating new content, the user is presented with the same textboxes but without selecting a course, lesson, or resource to edit. Instead, whatever the users adds to the textboxes will be saved as a new course, lesson, or resource when the user clicks the Save button. A screenshot of the Edit Lesson page can be seen in Figure 15.

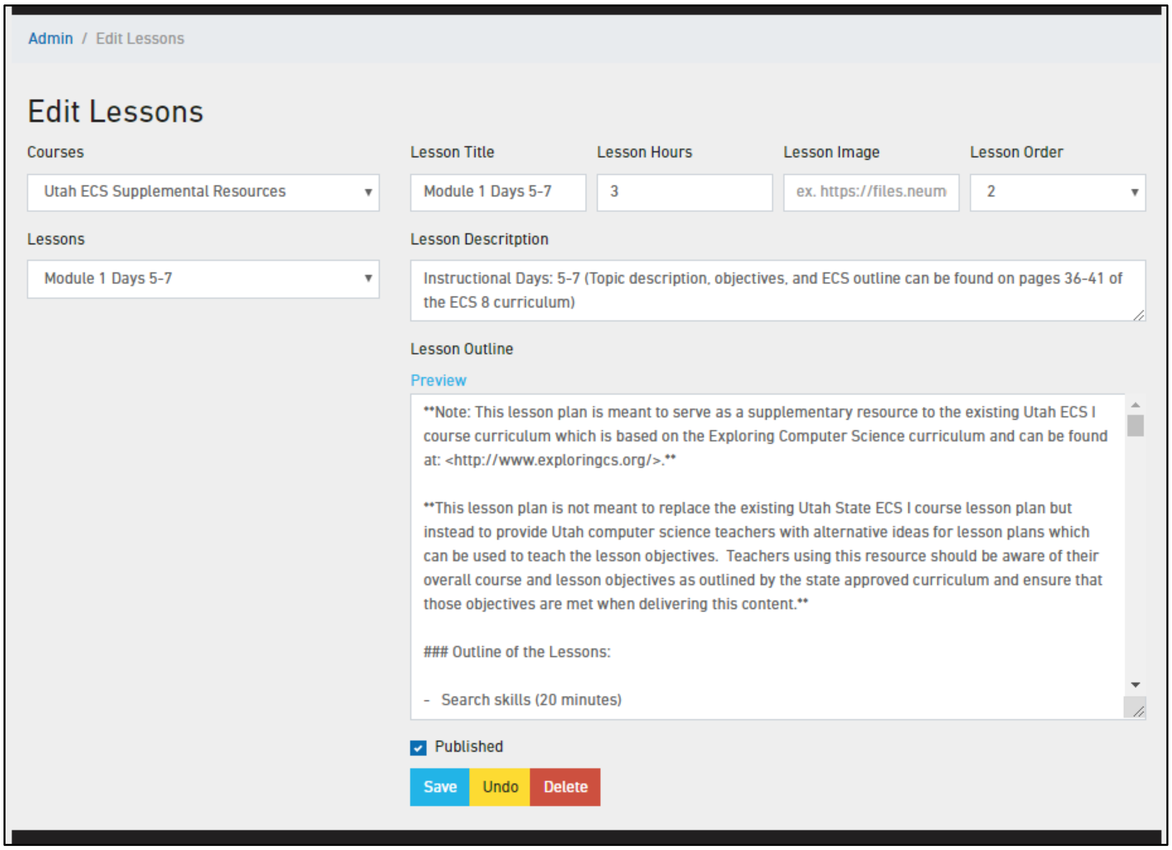


Figure 15 – Edit Lesson Page of the Portal

5.8 Summary

This section presented details regarding the portal website that is used to store the St. Owensby curriculum enhancements. The context and rationale for the portal website was established. A synopsis of the development methodology was provided. A summary of the initial system design was given as well as details regarding a subsequent revision to the system design and the user interface. Finally, screenshots of the final user interface for the portal were provided. The next chapter will focus on the results of the study including findings from the participant interviews and the survey respondents.

CHAPTER VI: RESULTS

The purpose of this study was to evaluate the effectiveness of collaborative online tools, the St. Owensby curriculum enhancements, and community partnerships in improving Utah K-12 computer science education. With the massive and growing gap between the number of annual computer science graduates and the number of open computer science jobs, improving interest in computer science as a major for the nation's college students is of paramount importance. By improving the effectiveness of computer science education in the nation's high schools, we may be able to positively impact the number of college students interested in majoring in computer science in the nation's colleges and universities.

The major assumption in this study is the premise that if high school computer science courses can be improved *and* if teachers can leverage an online platform to help them to gain content knowledge and better prepare to teach computer science courses without having to travel long distances, two things will happen. First, teachers will be more effective K-12 computer science teachers. Second, more students will exit high school with an interest in computer science as a college major choice and/or a career. Although this study depends heavily on this assumption, the assumption itself is not unfounded. In fact, administrators involved in the design and launch of ECS in Los Angeles found that the program positively impacted the computer science interests of the students. After taking ECS, the percentage of students who expressed an interest in continuing to study computer science increased from 17% to 43% (Goode, Chapman, & Margolis, 2012). The administrators also found that approximately 60% of the students reported an interest in working in the computer science field (Goode, Chapman, & Margolis, 2012).

Most of the research in K-12 computer science education revolves around ways to improve student learning. Although the researcher acknowledges the importance of student learning and the critical nature of continually improvement in that arena, this study did not focus on student learning. The researcher posits that understanding and improving the mindset, general preparedness, and self-efficacy of teachers is perhaps equally important as we strive to improve computer science education at the K-12 level. Hence, this research will focus on K-12 teachers rather than students.

6.1 Research Questions

The following research questions will guide this study:

1. What are the difficulties faced by Utah K-12 computer science teachers and how can technology tools and community partnerships be leveraged to alleviate some of those difficulties?
2. How are computer science teacher identity, self-efficacy, and enthusiasm of teachers impacted by using a collaborative curriculum portal developed through community partnerships?
3. How are computer science teacher identity, self-efficacy, and enthusiasm of teachers impacted by curriculum enhancements developed through community partnerships?
4. How does knowledge transfer through an interactive curriculum portal impact computer science teacher identity, self-efficacy, and agency in the classroom?

5. What are the perceptions of Utah high school computer science teachers and other key stakeholders regarding the impact of community partnerships on high school computer science courses?

6.2 Participants

Participants in this study were drawn from two distinct populations. For quantitative data, the survey found in Appendix B was emailed to all registered users of the collaborative curriculum portal. At the time of the study, there were 38 registered users. Of those 38 users, 23 of them (61%) responded by filling out the survey. A summary of the survey results will be presented in this chapter alongside results from the interviews. A report illustrating the full results of the survey can be found in Appendix G.

At the end of the survey, a question asked users if they would be willing to participate in an in-person interview with the researcher. Of the 23 responders, six indicated a willingness to be interviewed. The researcher conducted interviews with each of those six. Some of the interviews were conducted in-person and others were conducted over the phone. In both the in-person and the telephonic interviews, the researcher had the participant sign and return the consent form (found in Appendix D) prior to starting the interview. Interviews followed the format outlined in the interview template in Appendix A.

6.3 Interview Participant Demographics

Each of the six interview participants were K-12 teachers who teach the state-approved Exploring Computer Science course in Utah. Pseudonyms have been given for the participants to protect their anonymity. The average number of years of K-12 teaching experience among the interview participants was 17.6 with the high being 34 years and the low being four years. The participants averaged four years of teaching computer science with a high of eight years and a low of two. The participants had an average of 2.75 years of experience teaching Utah's Exploring Computer Science course with a high of four years and a low of one.

Four of the six participants taught ninth-grade students exclusively while the other two participants taught a mixture of 9^{th} through 12^{th} graders. Only one of the participants taught computer science exclusively. The other five participants also taught subjects such as film, media design, business, math, and foreign language. None of the participants had a computer science background prior to teaching. Their backgrounds include: photography, business, elementary education, and German.

A short description of the six interview participants is given in the next section of this chapter:

Ashley has been teaching K-12 education for six years, computer science for five years, and Utah's ECS I course for four years. Her academic background is in photography. She was told by her boss that she had to start teaching computer science in order to keep her job. She teaches film and media in addition to computer science. Ashley teaches 9^{th} through 12^{th} grade in ECS I.

Chase has been teaching K-12 education for five years and computer science and ECS I for three years each. His academic background is in business and he still teaches business classes in addition to computer science. Chase primarily teaches 9^{th} graders in the ECS I course.

Georgia has been teaching K-12 education for 34 years, computer science for eight years, and ECS I for one year. Her academic background is in business. She started teaching business information technology classes more than a decade before this study and helped write some of the state IT curriculum in business IT courses. She teaches 9^{th} through 12^{th} graders in the ECS I course.

Jason has been teaching K-12 education for 23 years and computer science and ECS I for four years each. His academic background is in business. He transitioned to teach computer science as a career decision because he felt that it was the best way to stay relevant and have long-term job security. He teaches only computer science courses and his ECS I class is comprised exclusively of 9th graders.

Kristina has been teaching K-12 for 22 years and computer science and ECS I for two years each. She was a business teacher for 20 years until two years ago when she started teaching computer science. She teaches ECS I to 9th graders.

MyKayla has been teaching K-12 for 15 years and computer science and ECS I for four years each. She has an academic background in mathematics and business. She was a business teacher and a math teacher until she started teaching ECS. She teaches ECS I to 10th through 12th graders.

The remainder of this chapter presents the qualitative findings from these interviews. Summary data from the quantitative survey will also be presented to support or contradict qualitative findings.

6.4 Difficulties Faced by Utah Computer Science Teachers

The interview participants unanimously spoke of a wide range of problems facing Utah's computer science teachers. Many of these problems were seen by the participants personally as well as demonstrated by their peers. The primary interview themes regarding difficulties faced are discussed in this section.

6.4.1 Many Computer Science Teachers Do Not Understand Computer Science

One of the biggest challenges faced by computer science teachers in Utah is a general lack of understanding of computer science principles which so many teachers face according to

Brandon Jacobson, an Information Technology Education Specialist at the Utah State Board of Education (personal communication, November 19, 2018). This lack of underlying computer science knowledge was evident in all six of the interview participants. Not a single participant had a computer science academic background of any kind. Additionally, several of the participants made the move to teach computer science because they were required to do so by their principal. Each of the participants in those circumstances described themselves as being at a disadvantage compared to peers who opted into the computer science courses on their own. They indicated that by being compelled to learn and teach computer science they had lower motivation, enthusiasm, and general interest in the subject matter.

Interview participant Ashley indicated that the lack of computer science knowledge hampered her understanding of the curriculum and her enthusiasm to teach it. She claimed that when she first began working on the computer science courses and started looking at the coursework, she "really did not like the curriculum." He continued, "I was resistant to the curriculum and mostly I think it was because I was afraid because I didn't understand it, I didn't know computer science." She posited that the reason she was ultimately successful was because she was able to trust that if she pushed forward, the curriculum would ultimately make sense. In her words:

> I knew enough to trust that I could see that the curriculum would make sense if I just trusted it. I was like, I know that smarter people than me wrote this curriculum so there's got to be a reason why they're doing it this way and, since I don't know the reason, I'm just going to have to trust it.

Ashely, who started teaching ECS I three years ago, was able to break through and understand the curriculum and the class as she developed a greater understanding of computer science. However, she fears that many of her peers get lost in the confusion and, failing to see the value of trusting the curriculum, struggle in teaching computer science effectively.

MyKayla, another interview participant, believes that the lack of subject matter understanding among computer science teachers creates a sense of fear and intimidation. She said that she sees peer teachers struggle with confidence in the classroom and even among their peers until they get over the hurdle of understanding basic computer science principles. She described the time when she was handed the computer science curriculum and, although there was training for the curriculum available to her, there was no training for the computer science concepts available. She was given a list of peers and instructed to contact them with questions but because of her lack of experience in computer science, she felt afraid to ask them for help out of a fear of looking like the only person who did not know what she was doing. She indicated that this happens with teachers across the state every year and, until they learn the basics of computer science, those teachers continue to feel vulnerable and scared in front of students and peers.

Brandon Jacobson of the State Board of Education admitted that he has also seen this phenomenon with teachers throughout the state. He attributed the teacher vulnerability and resistance to reaching out to peers for help to the notion that teachers feel they need to be the experts. He cited many of his business and marketing teachers who feel they need to be experts in the classroom and know all the answers for the students. As those teachers move into computer science classrooms, Jacobson struggles to get those teachers to understand that a) they will not know all the answers and b) that it is completely fine to not know all the answers and instead to approach the classroom with an attitude of joint exploration.

6.4.2 A Lack of Classroom-Ready Resources

A related theme that emerged in the interviews regarding difficulties faced by Utah's computer science teachers revolved around the premise that teachers struggle to find curriculum that can easily be adapted to their classroom. If teachers struggle to understand the computer science concepts, it might make sense that curriculum resources must be accompanied with detailed explanations and be classroom-ready. Each of the participants expressed concerns with the current state-endorsed ECS curriculum, claiming that it was out of date and that it failed to make topics interesting and relevant to students. Yet, according to the participants, teachers struggle to find other resources that they find valuable. Jacobson believes a large part of the problem is that teachers do not understand the computer science subject matter deeply enough to effectively modify existing curriculum resources for their own needs in the classroom (B. Jacobson, personal communication, November 19, 2018). Jacobson explained that:

> If [teachers] know computer science very well, and they find a lesson off the internet, they'll be able to pull out the main concepts, the main ideas, and flip it to something very personal to them and/or to their students and it makes that classroom even more engaging and the computer science principle much more impactful and permanent for the student. Whereas if they're still trying to figure out what they're supposed to be teaching or if they're leading from behind or learning with the students, they may struggle a little bit more with being creative because they don't see that complete connection between computer science and their personal situation or their students' personal situations (personal communication, November 19, 2018).

Interview participant Chase echoed similar sentiments. He cited concerns with the ECS curriculum, saying, "It's hard to tie in the ECS theories and the principles with the actual day to day use of computers." Chase further explained that he does not feel there is a dearth of resources available. Instead, he believes that the existing resources are not easily digestible and classroom ready. He claimed, "I get inundated with tech resources. The problem for me is definitely not the resources. It's resources that are quickly and easily accessible that I can understand and that don't require trainings to know how to use them."

MyKayla claimed that a lot of her peers have no idea how a computer works. She spoke about visiting training programs where teachers were learning for the first time what components make up a computer and how they interact. She claimed that was the primary reason why so many of her peers struggle to adapt lessons they find on the internet, they lack the understanding of underlying computer science concepts and are so far in over their heads that all they can do is repeat a lesson verbatim. She echoed the thoughts expressed by Brandon and Chase in citing an acute need for more customized, classroom ready resources.

Jason, another interview participant, also pointed to a need for more digestible curriculum resources. He pointed to the current ECS curriculum and its journal-heavy modules in which students are required to write down thoughts and lessons learned in daily or weekly journals. He lamented that he and many of his peers find the journaling to be difficult because it does not hold student interest and it decreases student interest and engagement in the classroom. However, he also claimed that because he lacks a fundamental knowledge in the computer science subject matter, he struggles to make changes to the curriculum. He believed there must be a better way to incorporate the learning outcomes achieved by journaling but still maintain student interest but also claimed that, "I'm not sure exactly what that would be."

6.4.3 Poor Collaboration among Computer Science Teachers

Several of the participants pointed to a lack of collaboration opportunities in computer science as a primary difficulty facing teachers in the state. This was typically associated with the fact that most computer science teachers are the only such teachers in their schools and they therefore lack peers with whom to discuss what worked and what did not work in each day's lessons. Several participants described an unwillingness to share and collaborate among some teachers. Georgia, one of the interview participants, pointed to her busy schedule as a reason for a lack of collaboration:

> I have an hour before school, and hour prep [class], and an hour after school, so I have three hours of time to prep and grade all of the papers. That's probably more than most people have. That being said [with all the classes I teach], that gives me 15 minutes per class to prepare.

Georgia believed that if teachers had more time to prepare for classes, they might have more time to help other teachers with their lessons and content.

However, some participants cited job security and the competitive nature of teaching as the reason for the lack of collaboration. MyKayla described the way she believes many of her peers feel regarding this competitive environment:

> You're competing with other people... So, if you're more fun than I am, or you're more engaging, then I'm not going to have as many students [as you], so I might not keep my job and you might keep yours. So, I might not share resources with you.

This lack of collaboration among computer science faculty further exacerbates the difficulties faced by these teachers. If a given teacher develops the understanding of computer science and becomes an effective teacher but lacks the time or desire to share with others, the potential positive impact of that teacher on the teaching community is reduced or eliminated.

6.5 Impact of the Curriculum Portal

Unanimously, all interview participants found that the curriculum portal was extremely easy to use and that it represented an improvement in curriculum accessibility. Chase felt that the portal made curriculum much easier to navigate than the state-approved ECS curriculum, which is made available to teachers in a single PDF file consisting of hundreds of pages. Georgia agreed, indicating that the way the portal is laid out allows teachers to easily drill down into a given lesson, read about it, and jump back to where they were before. Jason said that the curriculum portal is "a very easy resource to use" and that its ease of navigation made it easier for him to digest curriculum and understand it. Kristina and MyKayla also echoed similar sentiments. Both teachers found the portal extremely user-friendly and helpful. Kristina indicated that the way resources are available on the same page as lessons and you can click them and click back was extremely helpful to her. MyKayla said that she felt the portal laid out lessons and resources in a way which helped teachers easily identify more effective ways to engage their students.

Brandon Jacobson from the State Board of Education indicated that the curriculum portal has been well received from teachers across the state (personal communication, November 19, 2018). The teachers he works with have been sharing it with others and have echoed many of the positives mentioned by the interview participants (personal communication, November 19, 2018).

The data from the survey appears to validate the thoughts expressed by the interview participants. Of the 19 teachers who answered the question which asked if the curriculum portal presented lessons in a way which made the content easier to understand, 18 answered in the affirmative with 10 saying they strongly agreed and eight saying they somewhat agreed. Results from that question on the survey can be seen in Figure 16:

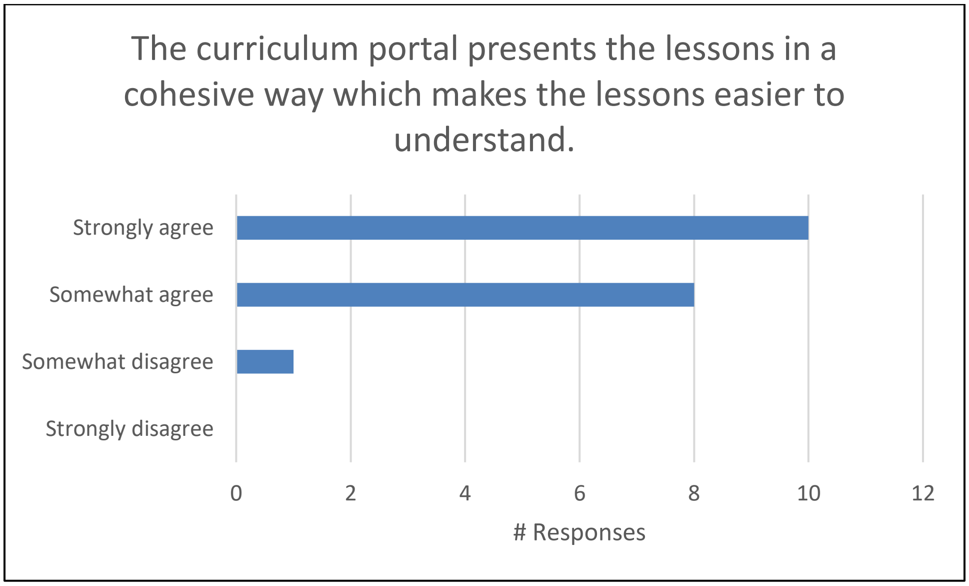


Figure 16 – Survey Results – Does the Portal Facilitate Curriculum Understanding?

One negative with the portal which will be discussed in more detail further on in this chapter has to do with the comment feature on each lesson. In that feature, teachers can share comments about lessons and provide advice to others. Each of the teachers indicated that they read comments, most of them found the comments valuable, but only one of them had ever created a comment. In fact, very few comments were created throughout the entire portal. In early September 2018, the administration at St. Owensby College created a contest to help inspire teachers to share their experiences with one another on the portal. In the contest, teachers could post comments on the portal throughout the month of September. At the end of September, two random comments would be selected and the teachers who created those comments would each receive a $100 gift card to Amazon.com. Information about the contest was sent to teachers throughout the state at the beginning of September.

Even though 38 teachers were actively using the portal, only four comments were created on the portal during the month of September as the contest took place. By December 2018, only 22 comments had been posted throughout the entire portal. MyKayla described this phenomenon and lamented that if teachers could get over their own reasons for not posting comments, the community of teachers would greatly benefit. She called those teachers who read comments but do not post themselves "lurkers" and cited that in national teaching organizations in which she participates, she sees the same kind of behavior. She claimed that "they just like to go in and get resources, but they don't want to post themselves." She primarily attributed that mentality to a lack of time, lack of confidence, or a fear of job security. More on those topics will be discussed in the knowledge transfer section of this chapter but it is ironic that MyKayla herself is a "lurker" on the portal, having never posted a comment. She appeared to have very high computer science self-efficacy, seemed to be doing very innovative things in her classroom, yet she felt that what she had to say would not have been of value. Her comments on the subject, which were: "why would I post, nobody will want to read it" highlight a disappointing self-conscious attitude that may be shared by many her peers based on the overall lack of sharing on the portal.

6.6 Impact of the St. Owensby Curriculum Enhancements

Each of the interview participants reported finding the St. Owensby curriculum enhancements to be beneficial. Ashley, who is in her third year teaching Utah's ECS I course, noted that the biggest benefit she found in the St. Owensby curriculum was an influx of new ideas on curriculum that hasn't changed significantly for some time. She specifically called out the lesson on binary (Appendix F, Module 2: Days 10-12) as a great addition to her classroom, saying that it helped the students who are visual learners more easily comprehend concepts of binary number systems. In the St. Owensby curriculum's binary lesson, teachers instruct the students on binary number systems and then use it in a practical way. The students take a 10x10 grid and create a rudimentary image by shading in each cell in some color. Then the students translate their picture to binary by converting each colored cell to a series of binary numbers representing 0-255 red, green, and blue (RGB) values. The students then pass the binary to a peer who then translates the binary back to color by shading in a 10x10 grid according to the color specified in the binary. The students then compare images to see if they created the same image. Ashley described the way this particular lesson impacted herself and her students:

> I'm a visual [learning] person and I know a lot of my students are more visual as well and the ones and zeros (of binary) don't really mean a lot to us, they don't mean anything to me, but the pictures [in the St. Owensby curriculum] and understanding that the ones and zeros become pictures is a whole new kind of exciting way to look at computers and images.

Georgia claimed that she used almost everything available in the St. Owensby curriculum and believed it helped her more effectively teach her students. Like Ashley, she claimed to be a visual learner and believed that a lot of her students learn the same way. She mentioned that a real strength of the St. Owensby curriculum is the heavy video content in many of the lessons:

> Most of the kids I see here are visual learners. To have a little video blurb that they can look at really helped them see what they were going to learn and what we were doing with it. They were fascinated with watching the videos.

Georgia also felt the St. Owensby curriculum did a better job of explaining concepts and teaching methods than the existing ECS curriculum. She described that often when reading the existing ECS curriculum she would be confused and the direction of the lesson and topics was unclear. She felt she could then turn to the St. Owensby curriculum and read through the lesson and really understand what the lessons was all about. She described feeling so dependent on the St. Owensby curriculum that when she ran through them all she felt stuck because there were no more to help her with the rest of her class.

Another interview participant, Chase, also felt a dependency on the St. Owensby curriculum. He described feeling "quite mad" when one of the St. Owensby curriculum lessons described the existing ECS content as being good enough and did not provide alternative ways to teach that lesson. Chase indicated that he believed students were more engaged in the class when he was following the lesson plans specified in the St. Owensby curriculum compared to the level of engagement in the typical ECS I lessons. He also said that the lessons helped him as an instructor be more effective because the St. Owensby curriculum described in greater detail what the teacher should do and think about compared to the state-approved ECS I curriculum. He felt that the way the St. Owensby curriculum provided direction to teachers "help [him] as a computer science principles teacher feel confident that I can teach the material" because the content is laid out in "a simple lesson plan that [he] can follow and [he] can make [his] own adjustments, but it has the information that [he] needs to teach the lesson successfully." He also indicated that he feels the St. Owensby curriculum works because they are directed at the teachers, not the students, unlike so many other available resources.

Interview participant Kristina sighted the freshness of the St. Owensby curriculum as a primary benefit to her. She felt that the enhancements provided updated content that she felt was severely needed in the state-approved curriculum. The fact that the enhancements focus on new topics such as current games or current trends and innovations in computer science helped her students get more engaged in the classroom because she felt the curriculum seemed more relevant to her students.

Jason also felt that the St. Owensby curriculum helped his students stay engaged in the course. He claimed, "[The St. Owensby curriculum] added some spice to my class. [The ECS national curriculum] was lacking. It wasn't really very exciting. The St. Owensby curriculum was a great success and added a good base and a lot of excitement." He said that he feels his class has been more engaged in the curriculum than in prior years and he pointed to the level of fun the students appeared to have been having in his classes with the St. Owensby curriculum. He indicated that, "School should be fun. This class should be fun" and he said that the enhancements made the class fun, which resulted in a better experience for him and for his students.

Overall, those who responded to the survey agreed with the interview participants. In one question, respondents were asked whether the St. Owensby curriculum helped them build interest in computer science among their students. Of the 18 respondents who answered that question, seven strongly agreed and 10 somewhat agreed that the curriculum did help them build computer science interest among their students. Results from that question are summarized in Figure 17.

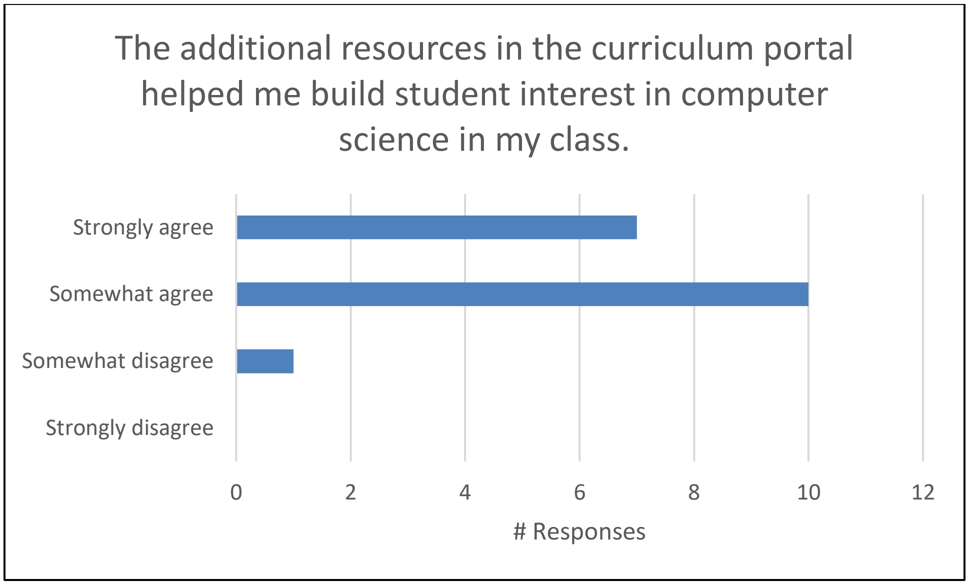


Figure 17 – Did the St. Owensby Curriculum Help Teachers Build Interest in Computer Science?

Likewise, survey respondents agreed with the interview participants on the topic of classroom engagement. One question on the survey asked if the St. Owensby curriculum helped them more effectively engage students in their class. Of the 18 responses, nine strongly agreed and eight somewhat agreed that the curriculum did help them engage students. Results from that question are summarized in Figure 18.

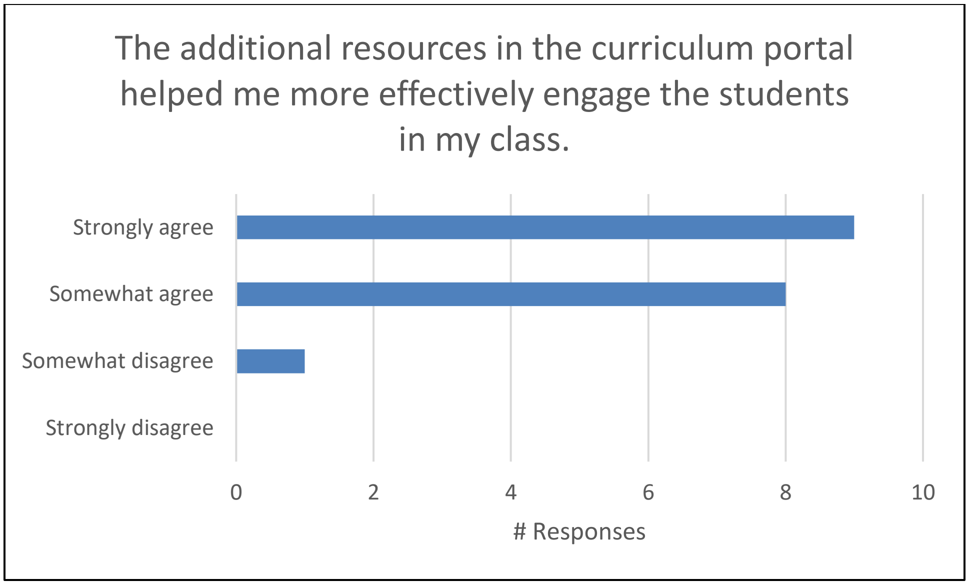


Figure 18 – Did the St. Owensby Curriculum Help Teachers Engage Students?

Finally, one question on the survey asked whether respondents would recommend the St. Owensby curriculum to other teachers. Of the 18 responses, 17 of them indicated they would recommend the curriculum to others. Results from that question are shown in Figure 19.

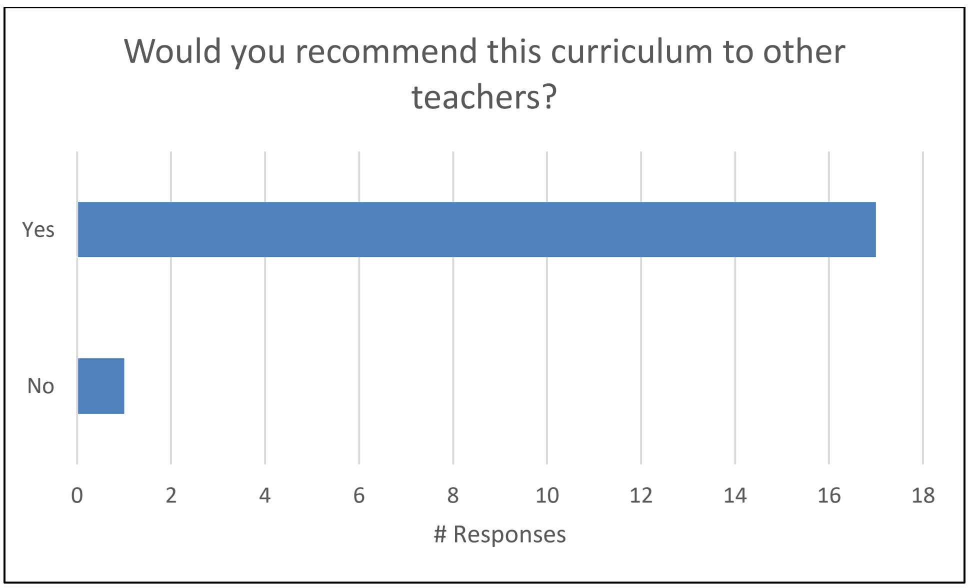


Figure 19 – Would Teachers Recommend the St. Owensby Curriculum to Others?

The positive impact of the St. Owensby curriculum cited by interview participants were wide-ranging and varied but the most common themes were that the lessons were easier to understand, and the content was more relevant and interesting to students. Two lessons in particular were brought up by participants multiple times as being exceptionally effective. Those were Human PacMan (Module 1: Days 15-16) and Fortnite Problem Solving (Module 2: Days 4-6). Plans for both lessons can be found in Appendix F. Given the frequency with which both lessons were praised, each will be discussed in greater detail in the following two sections. The researcher also visited several classrooms to observe both lessons being taught. Findings from those observations will also be presented.

6.7 Curriculum: PacMan

One of the challenges Jason and others have described regarding the current state-approved curriculum is that students often come into the class thinking they will dive right into programming. When instead the first two modules of the class are not about programming but rather focus on computer science topics and principles, the students lose interest in the class quickly, according to Jason. Subsequently, Jason has been searching for ways to improve the classroom engagement in his ECS I course. He found the St. Owensby curriculum added much-needed excitement to his class and did a better job of holding student interest than the state-approved curriculum. Particularly, he cited the PacMan lesson as being “a great success.”

The PacMan game takes place in Module 1, Days 15-16 of the St. Owensby curriculum. The topic description, found in the nationwide ECS curriculum states, “This lesson introduces the concept of a computer program within the context of a set of instructions for completing a common activity.” The ECS curriculum conveys the notion of following orders by using a directions quiz, some drawing activities, and a peanut butter and jelly sandwich exercise. The peanut butter and jelly sandwich exercise, which, according to many interview participants, has been a popular exercise among students, asks students to write instructions on how to make a peanut butter sandwich. The teacher then follows those instructions literally, the way a computer would interpret them, while using real peanut butter, jelly, bread, and a knife to make the sandwich. Because student instructions are often not extremely detailed, the results are often entertaining and provide a good lesson into how computers literally process instructions. However, feedback from some of the teachers being interviewed points to the lesson being a bit stale and in need of a refresh with current, more interesting content.

The PacMan lesson in the St. Owensby curriculum asks teachers to create a maze out of the desks and chairs in their classroom to recreate a life-size version of the game PacMan. They assign starting points throughout the maze for PacMan and for each of the ghosts and choose a student to play each of those roles. The game is run step-by-step where each character moves once and the game then pauses so the gameboard and players can be evaluated and discussed. Just as in PacMan, when a ghost captures PacMan, the game is over. The game is played a few times, alternating students in different roles. Then the students are asked to come up with rules for the ghosts and the PacMan in teams or individually. Rules can range from “always turn right” to “always move in the direction of PacMan” to something more complicated. The game is then replayed with PacMan and the ghosts interpreting rules created by the students. If the class is large enough, the lesson can turn into a contest to see who’s rules are most effective at capturing PacMan or eluding the ghosts.

When asked how Jason knew the PacMan lesson was successful in his class, he said, “That’s the feedback, anecdotally, that I got.” He went on to explain that in his class, he asks the principal to join the class for the day and act as PacMan. Jason said that the students have a great time and a lot of fun and that the principal felt the lesson was extremely successful. Specifically, Jason described the lesson’s benefits:

> It starts to get that idea of programming and what it takes. It has to be very precise, very exact, and the computer's only going to do what you tell it to do. [The lesson] is an especially good lead-in to programming.

Jason also really liked the PacMan lesson simply for the fact that it forced students to get out of their seat, be active, and participate. The researcher visited Jason's class in the fall semester of 2018 and noted similarly positive results. There were 38 students in the class and the room was already setup in a maze-like fashion, so no moving of desks was needed. The principal acted as PacMan and instantly, when students found out about the lesson and the role of the principal, the students appeared enthusiastic and excited about the lesson. Every one of the students participated in the lesson by either playing a character, creating rules, or both. Students seemed to be heavily engaged in the activity and invested in creating more effective rules than their peers. There was a lot of chatter on the sides of the room during the gameplay, but it appeared that the majority of the chatter was about the game, the lesson, the principal's role, and/or how to improve the rules. There was very little goofing off or distraction from the lesson. In talking to Jason about the lesson, he described that when using the peanut butter and jelly lesson, the class would typically have a small subset of students who were heavily engaged, others who were somewhat engaged, and about half the class who would not be interested or engaged. Jason felt the PacMan lesson helped his students better understand the concepts of how computers interpret instructions while at the same time doing a better job of maintaining student interest and increasing engagement.

6.8 Curriculum: Fortnite

Another lesson which was frequently praised by interview participants revolves around the popular video game Fortnite. The lesson takes place in Module 2, Days 4-6. The topic description in the corresponding lesson in the nationwide ECS curriculum states:

> Students will apply different strategies to help them make a plan and carry out the plan to solve several problems. These strategies may include (but are not limited to): draw a diagram or picture, make systematic lists, divide and conquer, find the pattern, and guess and check.

In the national ECS curriculum, the first four lessons, spanning the first nine days of Module 2 focus on the problem-solving process and how to implement it. Feedback from the teachers in the partnership created by St. Owensby was that the lessons were too stale and boring, and the content could be taught in much less time, leaving them to figure out what else to add to Module 2. The partnership introduced a new way of teaching students how to use the problem-solving method by having them dissect a popular video game and attempting to answer the question: "How did the video game developers make this game?" The lesson starts by showing a video of gameplay from the popular game Fortnite. The lesson also provides an alternate route, if teachers are uncomfortable with the game, by suggesting a scene from a movie (the 2018 Avengers: Infinity War) with heavy computer-generated imagery (CGI) which could be dissected with the same question just as effectively. The lesson then evolves into a discussion about "how did they do it?" with the teacher talking about scenes, components in scenes, rules and scripts for individual components, and even client-server character positioning. The lesson is full of individual or group work opportunities as students attempt to answer these questions themselves before a larger discussion. The lesson also contains detailed descriptions of problems and answers for teachers who may not fully understand the content themselves.

Kristina is one of the interview participants who praised the Fortnite lesson, saying, “the kids loved [it] because that’s what they play, it’s relevant to them.” She described the student’s reactions as they went through the lesson while dissecting a game with which they are familiar, “They’re like, ‘oh, oh! Okay, yeah, I get that now!’” She described that the students in her class were immediately interested in the subject and engaged because they love playing Fortnite. She said that when she played the Fortnite video at the beginning of class, every head lit up and students started critiquing the gameplay of the person in the video.

Brandon Jacobson from the State Board of Education explained that he has heard a lot of positive comments from teachers about the Fortnite lesson (personal communication, November 19, 2018). Jacobson pointed to teacher enthusiasm over student engagement as the main reason they love the lesson (personal communication, November 19, 2018). He also cited hearing various teachers praise the lesson for the way it explains the content to teachers, specifically calling out that the lesson explains to teachers at different points things like, “don’t worry if you aren’t sure about the answer just yet…” or, “if you don’t understand this concept yet, don’t be concerned, it will make sense as you read the rest of the lesson” (personal communication, November 19, 2018). Jacobson said that teachers have told him they want more lessons with that level of instruction and clarity for teachers (personal communication, November 19, 2018).

Interested to see the Fortnite lesson playout live, the researcher visited Kristina’s classroom in the fall semester of 2018. There were 24 students in the class and the students were engaged in the class start to finish. Students were laughing and talking to each other during the introduction video but the conversation all seemed to be around Fortnite and, mostly about the video in question so the crosstalk appeared to elevate the lesson rather than distract from it. The class was heavily discussion-based with some group assignments. There were at times so many comments that the teacher had to stop certain discussion points and move on to the next for the sake of completing the lesson. During the group work, teammates appeared engaged and interested in completing their assignment. Very few students appeared to not be completely engaged and no student appeared to be completely disengaged. Kristina compared this classroom environment to that which would take place when teaching the corresponding lesson in the national ECS curriculum by stating that she struggled to have any real discussion and engagement in the prior lessons and now, with the St. Owensby Fortnite lesson, she almost has too much of it.

6.9 Computer Science Teacher Identity

Teacher identity can be thought of as the degree to which a teacher identifies themselves as a teacher of a particular subject (Ni & Guzdial, 2012). Teacher identity has been found to be a major factor directly related to teaching quality and teacher commitment (Cardelle-Elawar, Irwin, & Sanz de Acedo-Lizarraga, 2007; Chan, Lau, Nie, Lim, & Hogan, 2008). In this section, computer science teacher identity concerns expressed by the interview participants will be discussed as well as the perceptions of the impact of the portal and the St. Owensby curriculum on teacher identity.

6.9.1 Computer Science Teacher Identity Concerns

One of the biggest concerns regarding teacher identity that was expressed in the interviews was the notion that many computer science teachers in Utah are required to teach computer science to keep their jobs rather than opting into the field out of passion or interest. Interview participants Ashley, Kristina, and MyKayla all fit that demographic. MyKayla referred to the phenomenon as being "voluntold" to teach computer science. She claimed that being "voluntold" had a severely negative impact on her computer science teaching identity and that of her peers in similar situations. She described feeling anxious about computer science, not understanding it herself and wondering how she was supposed to teach it. She felt that even though she was directed to numerous teaching resources, she felt there was a lack of resources available because she did not understand any of the resources to which she was directed. She claimed that not only did this affect her identity as a computer science teacher, it impacted her identity as a teacher in general as she wondered if she could ever be effective in helping students understand the concepts of her classes.

MyKayla also described a concern that teachers who teach multiple other subjects may not identify as a computer science teacher because they are spread so thinly across multiple disciplines. This may be compounded when those teachers do not have a computer science academic background, which was the case for each of the six interview participants. She described peers who she knows claim they are "a business teacher who teaches one section of computer science" or "a math teacher who teaches a little bit of programming."

Interview participant Jason expressed an opinion that low computer science teaching identity negatively affects performance. He felt that teachers who identify as non-computer science teachers would likely gravitate to the subject with which they identify and look for ways to evolve lessons toward principles in those subjects rather than sticking to the core computer science curriculum. He cited one business teacher he knows who treads as lightly as possible on the programming concepts and pushes his class to research more into ways computers are used in business because that domain is familiar to him. Jason posited that such a slant on the course content can represent great lessons for students but expressed concern that significant deviations from the course objectives may cause more harm than good.

MyKayla agreed that identity issues can limit the effectiveness of teachers. She pointed to lessons which teach students about the lucrative, exciting, and plentiful computer science careers and posited that if teachers don't identify as a computer science teacher, they likely don't fully understand those careers and therefore likely have a difficult time helping students understand them. She also pointed out her belief that in order to be fully effective in teaching, a teacher must understand the course content to the point where they can customize it to their classroom and teaching style but teachers with low computer science teaching identity may not be able to do that. She felt that it is difficult for students to relate to existing curricula, including the national ECS curriculum, and that such customization is needed in computer science perhaps more than in other disciplines. But she reiterated that those who lack the computer science teaching identity will likely struggle making those customizations to their coursework.

6.9.2 Impact on Computer Science Teacher Identity of the Portal and the St. Owensby Curriculum

Interview participant Jason expressed an opinion that the portal has a positive impact on computer science teacher identity in computer science. He cited that the portal is housed on St. Owensby servers and accessed via a St. Owensby URL as helping teachers because those teachers know they are going to a college website and that the college focuses on computer science education, adding legitimacy both in subject matter and academics to the portal content. He also claimed that the portal impacted his computer science teaching identity because the layout was so simple and easy to use, for the first time he felt that he had all the resources he needed at his fingertips citing, "it gives me a chance to be able to pick and choose what I feel comfortable with, and what I think would benefit the students."

Kristina agreed with Jason on the positive impact the portal has on computer science teaching identify. She felt that the more information she has and the more comfortable she is with that information, the more she feels like she can teach that information. She felt the portal and the way the curriculum and resources are laid out and outlined helped facilitate that information comprehension in ways that existing resources do not. Chase echoed similar sentiments, claiming that the portal is "easy to navigate and easy to get the resources that I need to feel more competent. In that sense, it helps me with my motivation and identity as a computer science teacher."

Georgia felt the St. Owensby curriculum significantly improved her identity as a computer science teacher because of the way the lessons explain topics to teachers. She cited feeling confused reading existing curriculum resources and then obtaining a greater understanding of the lessons and content when reading through the St. Owensby curriculum. She believed that directly impacts her computer science teaching identity because it reinforces to her that she can succeed in teaching the subject effectively. Chase agreed, citing that the St. Owensby curriculum builds his computer science teaching identity because "it helps me feel like I can teach the lessons." He also pointed to the St. Owensby curriculum and the way the content not only presents ideas to discuss with students but tips and insight for teachers regarding the best ways to present different subjects. He praised the layout of the St. Owensby lessons, saying each topic is laid out in a, "simple lesson plan that I can follow, and I can make my own adjustments, but it has the information that I need to teach the lesson successfully." He reiterated that such a convenient layout helps develop his sense of computer science teaching identity.

6.10 Computer Science Self-Efficacy

Several studies throughout the nation have shown that high school computer science teachers often struggle with a variety of self-efficacy-related issues (Cutts, Robertson, Donaldson, & O'Donnell, 2017; Diethelm, Hildebrandt, & Krekeler, 2009; Yadav, Gretter, & Hambrusch, 2015). Teachers' lack of confidence in their ability to teach computer science topics can negatively impact their ability to teach effectively (Cutts, Robertson, Donaldson, & O'Donnell, 2017). Many computer science teachers are experts in other subjects such as mathematics or business and learned computer science at the request of their administrators (Cutts, Robertson, Donaldson, & O'Donnell, 2017). This was the case with all six of the interview participants in this study. The interview participants had backgrounds in mathematics, business, foreign language, film and art, and other subjects outside of computer science. In this section, computer science self-efficacy concerns shared by interview participants are presented as well as their perceptions regarding the impact of the portal and the St. Owensby curriculum on teacher self-efficacy.

6.10.1 Computer Science Self-Efficacy Concerns

Interview participant Georgia admitted she sees computer science self-efficacy issues in her peers who are new to teaching computer science. She posited that a lot of teachers in computer science do not understand how computers work and do not grasp basic computer science principles. She believes that lack of understanding has a negative impact on their self-efficacy in the subject. Ashley agreed with Georgia, citing that new computer science teachers struggle because they typically come from other academic backgrounds and they do not understand computer science. She admitted those teachers are struggling and they "are going to continue to struggle" until they finally break through that understanding. She recalled providing advice to a peer teacher who was new to computer science, telling him, “You’re going to struggle in the beginning, it’s going to be awkward at first because you’re not used to thinking this way [like the way computers process information], but you have to work at it and finally you’ll get it and you’re going to love it.”

Georgia believed that the computer science self-efficacy issues are compounded by a mode of thinking where teachers feel they have to know all the answers in their classrooms, saying, “I think teachers need to quit thinking that they’re a failure if they don’t know everything.” Kristina also admitted to seeing this as an issue which contributes to poor computer science self-efficacy. She posited that the notion that teachers need to know everything about their subjects is a societal norm that creates an unhealthy expectation for teachers, particularly those teaching subjects with which they have limited familiarity. She described her belief that the teacher “is not the be-all, end-all” and should not know all the answers all the time. Instead, she tries to get the students working together to find answers to questions before they ask her the question directly. She indicated that she believes working out questions on their own reinforces learning at a higher level while also removing the burden from her to know all the answers. However, she also pointed out that she has had issues with parents of students who object to this form of teaching and believe the teacher should know all the answers and be the primary source of those answers for their students. She claimed, “I've had a lot of parents who got mad at me because I made the kids work with each other before they asked me questions.” Such a reaction by parents may reinforce Kristina’s position that a societal norm exists through which the expectation is that teachers should know all the answers about subjects they teach. Kristina claimed that many of her peers suffer from computer science self-efficacy issues because of fear associated with this expectation. She said that many teachers she works with exacerbate their own self-efficacy issues with questions such as, “What do I do if [the students] ask me a question and I don’t know the answer?”

Finally, Georgia related the computer science self-efficacy issues back to curriculum concerns discussed previously. She claimed that if teachers struggle with self-efficacy, it affects their ability to “own” their curriculum and customize it to their needs. As a result, curriculum resources need to very adequately and clearly spell everything out to a teacher and the more classroom-ready a resource is, the more valuable it will be to the teaching community.

6.10.2 Impact on Computer Science Self-Efficacy of the Portal and the St. Owensby Curriculum

Interview participant Chase described his own computer science self-efficacy issues when he first started teaching ECS I, stating:

> I just remember the first time I taught [ECS I], I had gone to the training, and I remembered the concepts from the training and felt I was prepared for the lesson and I started giving the lesson and I realized I didn't know all the answers. I couldn't quickly locate answers, and it was not a fun situation to be in.

Chase went on to explain that the way the St. Owensby curriculum provides direct instruction to teachers about specific points of discussion really helped him through that self-doubt and concern, building his confidence along the way. Speaking of both the layout of the curriculum and the portal, Chase continued:

> My biggest worry when I started teaching computer science was that I wouldn't know how to teach it because I hadn't taken a lot of computer science college classes and such. When I try to teach a lesson on some concept that I barely know anything about, I would find myself googling and YouTube-ing[sic] and just trying to find anything to become

> more educated on it. Having a layout like [the portal and the St. Owensby curriculum] removes the need to be searching all the information and instead, in 10 minutes I can review anything that I need to brush up on. I think that's a big factor.

MyKayla also felt that the layout of the curriculum and portal improved computer science self-efficacy. She posited that many teachers who start out in computer science do not understand basic computer science principles. She explained that if teachers see the lesson modeled in front of them through a well-explained lesson plan, they can visualize it and it can reinforce their confidence that they can be successful in this subject matter.

Brandon Jacobson from the Utah State Board of Education also felt that the layout of the curriculum can improve computer science self-efficacy. Brandon explained that he has heard from several teachers that the way the St. Owensby curriculum lays out topics by suggesting a timeframe for each discussion point and activity helps take some of the guessing out of building a lesson plan (personal communication, November 19, 2018). He explained that something as simple as identifying how long a course or activity within a course should take is "something that [our teachers] always struggle with" and that struggle is related to their lack of subject matter expertise (personal communication, November 19, 2018).

6.11 Enthusiasm and Motivation

Another factor that directly impacts a teacher's efficacy is their intrinsic motivation. Eccles and Wigfield (2002) found a strong correlation between intrinsic motivation of teachers and students' ability to adapt and learn. Teacher's intrinsic motivation is closely related to teacher enthusiasm and can be measured by the degree of positive experience teachers encounter while teaching a given topic (Kunter, Frenzel, Nagy, Baumert, & Pekrun, 2011). Long and Hoy (2006) found that teacher enthusiasm is highly correlated with effective instruction and can directly impact the motivation of students. In the rest of this section, concerns regarding teacher enthusiasm as revealed by interview participants will be presented followed by interview participant perceptions of the impact of the portal and the St. Owensby curriculum on teacher enthusiasm.

6.11.1 Enthusiasm Concerns

Interview participant Georgia posited that teacher enthusiasm directly impacts teacher success. She claimed to have seen this with her own students where, upon reflecting on her teaching, she finds, "I'm not passionate about [the subject], and the kids have picked up on that." She indicated that she has seen the same phenomenon among other teachers with whom she works both in computer science and in other disciplines. She recalled a conversation with a teaching assistant where they were trying to identify why the students were struggling to pick up HTML concepts. Ultimately, Georgia realized that, "HTML is not my favorite thing to do" and she noticed that the students had learned the same attitude toward HTML from her.

She also pointed to teachers who are forced to go into computer science, positing that they likely have lower enthusiasm than their peers. She said:

> I think that with teachers who are forced to teach something they don't want to, it's hard to have that passion for that subject and [having that passion] is half the battle. If you're having fun doing something, then the kids are going to have fun doing it.

Chase also cited that he has worked with teachers who lack the passion and enthusiasm for computer science teaching and he believed that those teachers struggle to get students to understand the course content. Conversely, pointing to teachers he believed to be very effective at teaching computer science, he said, "I think they definitely wouldn't have been as effective if they didn't enjoy it." Both Georgia and Chase indicated that teachers who lack understanding of computer science likely have decreased enthusiasm for the subject matter and therefore likely have lower effectiveness compared to their peers.

6.11.2 Impact on Enthusiasm of the Portal and the St. Owensby Curriculum

Chase felt that the portal and its easy-to-navigate interface directly impacted his motivation and enthusiasm to teach computer science. He said the portal "is just easy to navigate and easy to get the resources I need to feel more confident. In that sense, the portal helps me with my motivation and identity as a computer science teacher." Jason agreed with Chase, citing that he felt lost in the state-approved ECS curriculum but that the organization of the portal where the curriculum and resources are easily navigable and found in the same locations as one another got him more excited about teaching computer science in the ECS I course.

Ashley felt the St. Owensby curriculum helped her feel more enthusiastic about teaching her computer science courses. She described herself as a teacher who is always changing her teaching methods and looking to improve her curriculum and delivery. She said she never teaches the same lesson the same way and is always looking for new ways to approach a given subject. She summarizes her enthusiasm for the portal: "it's kind of fun to think about what I would do different to keep myself engaged and to teach [ECS I] better so I'm always improving as I go. That's kind of why I think it's so valuable to have a portal like this."

Georgia also felt the St. Owensby curriculum helped her with her enthusiasm. She pointed to the notion that teachers get positive reinforcement when lessons go well, saying, "Well, I love anytime a lesson goes great, then you're super-excited to teach it again the next time. But then you have days when they don't go well, and in those times you have to motivate yourself to teach again. But anytime these kids are excited, so am I." She described a couple of the times when she implemented St. Owensby lessons, "the kids, they just had a blast. I had kids staying after school because they didn't finish, and they were going to make sure they finish the lesson." She reiterated that when lessons go well, it builds enthusiasm. She even mentioned to the students that she was trying new ways to teach the ECS I course by using the lessons from St. Owensby. She claimed that when certain St. Owensby lessons were finished, students were asking for more: "the kids, they love [the St. Owensby lessons] and they ask, 'Do you have more of those lessons? Do you have different ones? Do you have any more?'" Georgia mentioned that when she has that kind of positive reinforcement from students, it helps cement her love of computer science and her love of teaching.

Finally, Kristina also mentioned that the curriculum and the portal helped build her enthusiasm for the course content. She described herself as a curriculum user, not a curriculum developer. She believed she lacked the foundational understanding of computer science to effectively change and modify the lessons so she relies on others to build the curriculum in a way which she can use. According to Kristina, the portal and the St. Owensby curriculum provided her with a wide range of resources and enough instruction to effectively give her a lot of options for her class. In her mind, that deeper understanding and added flexibility translated to higher enthusiasm for the ECS I course.

6.12 Knowledge Transfer

In 2012, Ni and Guzdial from the School of Interactive Computing at Georgia Institute of Technology launched a study to identify characteristics of high school computer science teachers. The authors used semi-structured interviews to collect qualitative data on nine high school introductory computer science teachers (Ni & Guzdial, 2012). Every teacher who participated in the study claimed to have felt isolated and unable to locate adequate sources for the training they required in computer science education (Ni & Guzdial, 2012). The findings of

Ni and Guzdial underscore the difficult situation in which many computer science teachers find themselves as they seek to teach difficult concepts without proper training and access to resources. This scarce access to training and collaborative opportunities severely limited the amount of knowledge transfer taking place among computer science faculty education (Ni & Guzdial, 2012). The lack of knowledge transfer causes two negative effects: (a) new or inexperienced teachers lack the training and mentoring needed to be effective; and (b) the benefits of having an exceptional teacher are diminished because that teacher's strong teaching skills are not adequately spread to his or her peers.

A lack of sufficient training and access to collaboration opportunities among computer science faculty were also found to be issues affecting Utah's high school computer science teachers, according to Jed Bodily, Manager of High School Relationships at St. Owensby College (personal communication, September 1, 2018). Brandon Jacobson at the State Board of Education also cited the fact that teachers are often isolated, being the only computer science teacher in their school, and hence have limited collaboration and training opportunities (personal communication, November 19, 2018). Each of the six interview participants also shared concerns regarding training and collaboration among computer science teachers. Many of them also pointed to such knowledge transfer opportunities as being critical to their personal development and that of their peers. The most common themes regarding knowledge transfer concerns that arose in the interviews will be discussed in this section as well as the perceptions of the participants on the impact of the portal on teacher knowledge transfer.

6.12.1 Knowledge Transfer Concerns

All interview participants admitted the lack of training, collaboration, and the resulting scarcity of knowledge transfer negatively affects teachers throughout the state. Chase felt that the primary cause for the lack of knowledge transfer was isolation among computer science teachers, particularly for those who are the lone computer science teacher in their school. He believed that the majority of computer science teachers in the state are in that situation, with few or no peers within their school from whom they can solicit ideas or with whom they can collaborate. Ashley, Georgia, and Jason all posited that in-person trainings are the preferred method of learning among teachers and each of them pointed out the vast number of Utah teachers who work and live in rural areas, making in-person training attendance difficult.

Georgia indicated that she thought more teachers could start knowledge transfer through trainings, mentoring, or collaboration if they would just ask. She cited that the teaching community is typically open and helpful to one another and requests for help would typically not go unanswered. However, she felt that many teachers do not ask for help because they feel they need to know the details of their subject and asking for help makes them appear vulnerable and weak. She said, “I think some teachers, they think it makes them look stupid if they go ask somebody else to help. I think that's something we need to work on.”

MyKayla claimed one of the contributing factors to the lack of knowledge transfer was that there was no central training authority for computer science teacher education in the state. She pointed out that training is conducted and overseen at the district level and that many districts in the state have no computer science teaching expert themselves and hence, the training suffers from inadequacy. She claimed to have known of districts where there was no training and no resources provided to teachers because the district leadership themselves did not know about the various resources or how to use them. This abandonment of localized training caused further feelings of inadequacy and isolation for the computer science teachers in those districts, according to MyKayla.

Finally, MyKayla also admitted to seeing administrators pushing back on certain types of training for computer science teachers out of fear that those teachers might leave for better paying jobs once fully trained on computer science topics. She indicated that she has heard this "train and leave" attitude mentioned among her peers and she feels some teachers approach training with the intent of learning as much as possible in order to qualify themselves for work in the higher-paying computer science industry. MyKayla pointed out the difficult dilemma facing high school administrators who need their teachers to be trained but do not want them leaving for other positions. She posited that having teachers and administrators at odds with each other regarding levels of training needed and reasons for that training likely further decreases computer science teacher self-efficacy and enthusiasm.

6.12.2 Impact on Knowledge Transfer of the Portal

Ashley indicated that the portal was very valuable in terms of knowledge transfer. She claimed as she went through the portal looking for curriculum, she noticed the comments from some of her peers and she said, "I was kind of looking for some new stuff on [a particular lesson from ECS I] or some different things I could do and [the portal] gave me some good ideas." She mentioned having found resources that she had never heard of through comments from peers on the portal. She also claimed to have sent an email to some teachers in her district, sharing thoughts from the comments on the portal. She continued, describing her thoughts on the portal as a tool for knowledge transfer, saying, "I think it is very valuable."

Chase also found the portal and the comment feature to be valuable in facilitating knowledge transfer. He claimed to particularly like the way the comments are laid out along with the lesson itself so everything related to the lesson can be found in the same place, "I do look at the comments and I think they're easy to read and easy to see because they are all linked together really well." Chase related the ease-of-use of the portal to the ongoing need to have more classroom-ready resources for teachers. The less work a teacher has to do to piece together parts of a lesson, the more time they will have to absorb the content and plan for preparing the lesson in their own classroom, according to Chase. Regarding the portal as a tool for knowledge transfer, he said:

> I think that the more detailed and specific the course content is, the better. [In the portal], everything on a given page is specifically for that lesson. It's super convenient. If I was to get an email where somebody gave some advice for module 25 of such and such, I'd probably just delete it and never think about it again. But in the portal, where it's right there as I'm looking at the lesson, I can see any comments or other information that makes it helpful just for that lesson, which I like.

Each of the other interview participants also claimed the portal can facilitate knowledge transfer through the comment feature. However, every one of the participants lamented that so few people place comments in the portal to create that knowledge transfer. The participants' concern over the lack of comments is somewhat ironic because they were self-reported beneficiaries of knowledge shared by peers through the comment sections but they themselves did not reciprocate. Despite seeing the benefit they were receiving by reading comments from other teachers on the portal, none of the interview candidates ever placed a comment themselves on the portal.

In early September 2018, the administration at St. Owensby College noticed the low participation from teachers on the comment section. In an attempt to encourage teachers to share their experiences and knowledge in the portal, St. Owensby College administration created a contest on the portal. In the contest, teachers could post comments on the portal throughout the month of September. At the end of September, two random comments would be selected and the teachers who created those comments would each receive a $100 gift card to Amazon.com. Information about the contest was sent to teachers throughout the state at the beginning of September. Chase mentioned that he had seen the note informing teachers of the contest but still did not place his experiences or knowledge on the portal in the form of a comment. MyKayla did the same. She is a member of the national Computer Science Teachers Association (CSTA) and she mentioned seeing the same sort of behavior with that group. A national CSTA drawing was conducted on a CSTA website where, if teachers placed ideas for curriculum and coursework on the website, they would have a chance to win a $50 prize. MyKayla said that only one person participated nationwide. Reasons why the interview participants say teachers are reluctant to share ideas are presented in the next section.

6.12.3 Reasons for Not Sharing Ideas on the Portal

Interview participants Chase and Jason pointed to time as the primary factor why teachers do not share ideas with others. Chase said:

> My biggest, and I think every teacher's biggest, concern or problem, I guess, is time. I have lots of different preps with my various classes. I think that the sharing opportunities are there a lot more than I use them just because I don't have time.

Citing concerns over his workload and the number of classes he has to prep for each day, Chase continued:

> I don't want to take my five minutes of my time to put up a comment. I want to take my five minutes to get my next test ready or my next class ready. It comes down to prioritizing my time. I benefit from others' comments, and I'm sure others could benefit from my comments. I just don't take the time.

Jason cited that "time is the biggest factor" regarding why he did not add his experiences or knowledge to the portal. He said, "As you probably well know, teachers put in a lot more time than people think, so I think time is the biggest reason people don't add their expertise to the portal."

Georgia believed that the biggest factor regarding why teachers do not share expertise with others is a lack of self-confidence or self-efficacy in computer science. She indicated that she has seen teachers who will not ask for help because they think it will make them look like a failure. She said that the same mindset prevents teachers from sharing their ideas, thinking that perhaps other teachers will not value their perspectives or ideas, further reducing their own self-confidence. Kristina felt the same way, citing that teachers likely do not contribute to sharing groups because they lack the self-confidence in their own teaching. She believed teachers think to themselves, "Oh, what I have to say isn't valuable." She admitted, however, that "as teachers, we're always so hard on ourselves" and that if teachers could break out of that mindset, it would be positive for all involved.

MyKayla admitted to feeling self-conscious herself about posting her expertise online. She believed that if others read her comments, they would simply say, "oh, that's just MyKayla, she does things in a weird way." Chase also felt that self-confidence is an issue with teachers when it comes to sharing knowledge. He cited that many computer science teachers have self-confidence issues with their own course and their own students for a variety of reasons. To Chase, it only makes sense that they would have even more concern about getting up in front of their peers to share ideas. Jason agreed that self-confidence is a contributing factor in teacher reluctance to share ideas. Yet Jason also admitted that most teachers know that their lack of sharing is a bad mentality to have. He believes that teachers are so overwhelmed with time and self-confidence issues that they simply say to themselves, “I’m just going to get through this and maybe next year it’ll change.”

Jason also identified some portion of the teaching population that he believed is adverse to sharing ideas for job security reasons. He claimed that, “sometimes teachers are a little bit proprietary. In other words, they don’t like to share their ideas.” MyKayla also indicated that she believed teachers do not share out of job security. She claimed that some teachers view other teachers in their school or district as vying for their jobs and then they develop the mentality of, “if you're more fun than I am, or you're more engaging, then I'm not going to have as many students, so I might not keep my job and you might keep yours. So, I might not share resources with you.” Chase echoed similar thoughts adding that, not only are teachers going to be making their competition better by sharing, but, he said, there are, “no incentives to share with peers other than just being a good collaborator.”

6.12.4 Ideas for Improving Knowledge Transfer

Interestingly, when asked how to improve collaboration among teachers, those who had ideas among the interview participants all pointed toward in-person training. Kristina recommended, “I would say the best vehicle is the summer conferences. Teachers are looking for new ideas, things to do. And answers to the question, ‘How do I teach that?’” She indicated that for the conferences you can identify great teachers and tell them they have to present their ideas and their teaching methods. She felt that teachers attend summer conferences every year to find ways to improve their teaching.

MyKayla also thought the conferences would be the best place for knowledge transfer. She pointed out that when sharing with others, “some teachers might be nervous” but then claimed that perhaps, “If you had a teacher with a couple teachers, or a group of teachers, or a workshop” at the summer conference, the teachers would get past their sharing anxiety.

Interview participant Jason also felt conferences are the preferred way of sharing between teachers. He believed that the act of attending a conference put you in a mindset where you are ready to learn, share, and collaborate:

> I enjoy going to the [summer] conference and meeting people, getting to know the people, and have ideas shared. I think at the conferences people are usually pretty willing to share. If they're there, I think they've already got that mindset that maybe I'm going to share, or somebody's going to share some information with me. I think they're a little more open to that than someone who wouldn't go.

None of the interview participants recommended anything regarding the portal or other online systems for improving collaboration among teachers and facilitating knowledge transfer. Each of the responses were related to the Utah summer teaching conference and how to improve teacher idea-sharing at that venue.

6.13 Summary

This chapter presented findings from a study in which data from six interview participants and 23 survey respondents were collected and analyzed. The findings show that computer science teachers face tremendous challenges in the state of Utah. One of the biggest challenges is that teachers do not understand computer science principles, mainly due to the fact that most teachers are not computer science professionals or did not study computer science academically. Another challenge teachers face is the lack of classroom-ready resources available for their classes. With a general lack of understanding of the subject matter, the lack of classroom-ready resources causes more strain as teachers are forced to digest and modify lessons before presenting them to their students. The third major challenge found in this study is that teachers lack collaborative opportunities to share with and learn from peers. This is often because many computer science teachers are the only such teachers in their school but is also related to distance from training opportunities in rural areas and lack of time to collaborate.

The curriculum portal and the St. Owensby curriculum enhancements were extremely well-received by the interview participants and the survey respondents. Nearly all participants felt the portal and the St. Owensby curriculum had a positive impact on computer science teacher identity by providing easier to digest resources in a more classroom-ready and easily navigable way. Likewise, nearly all participants agreed that the portal and the St. Owensby curriculum had a positive impact on computer science teacher identity. By providing resources that are beneficial to teachers and empowering to them in their classrooms, teacher self-efficacy and, hence, identity, are magnified. The same impact was found on teacher enthusiasm as well. As teachers find their classrooms running more smoothly and the curriculum making more sense to them, they gain enthusiasm for teaching the subject matter, as was pointed out by each of the interview participants.

The portal was found to have had a positive impact on knowledge transfer by the participants. However, the dearth of activity in the portal's comments section lead to further questioning of the participants regarding their lack of willingness to share ideas with peers on the portal. The teacher participants identified that major factors in reducing collaboration among teachers are: lack of time, lack of computer science self-confidence or self-efficacy, and lack of job security. The participants felt that teacher conferences are the preferred method of facilitating knowledge transfer among faculty.

CHAPTER VII: DISCUSSION

This chapter presents a discussion of the findings of the study. The purpose of this study was to evaluate the effectiveness of collaborative online tools, curriculum enhancements, and community partnerships in improving Utah K-12 computer science education. With the massive and growing gap between the number of annual computer science graduates and the number of open computer science jobs, improving interest in computer science as a major for the nation's college students is of paramount importance. By improving the effectiveness of computer science education in the nation's high schools, we may be able to positively impact the number of college students interested in majoring in computer science in the nation's colleges and universities.

The following research questions guided this study:

1. What are the difficulties faced by Utah K-12 computer science teachers and how can technology tools and community partnerships be leveraged to alleviate some of those difficulties?
2. How are computer science teacher identity, self-efficacy, and enthusiasm of teachers impacted by using a collaborative curriculum portal developed through community partnerships?
3. How are computer science teacher identity, self-efficacy, and enthusiasm of teachers impacted by curriculum enhancements developed through community partnerships?

4. How does knowledge transfer through an interactive curriculum portal impact computer science teacher identity, self-efficacy, and agency in the classroom?
5. What are the perceptions of Utah high school computer science teachers and other key stakeholders regarding the impact of community partnerships on high school computer science courses?

7.1 Discussion of Findings

This section will present a discussion of the findings, ordered by research question.

7.1.1 RQ 1: Difficulties Faced by Utah K-12 Computer Science Teachers

The primary difficulties cited by the participants in this study pointed were that teachers lack an understanding of computer science fundamentals, the dearth of classroom-ready resources, and the limited availability of collaboration opportunities. Each of those difficulties will be discussed in the rest of this section.

7.1.1.1 Lack of subject matter understanding. The existing research identified the lack of subject matter expertise as a considerable challenge to K-12 computer science teachers nationwide. In 2015, the Computer Science Teachers Association (CSTA) pointed out that the number of computer science-related teacher certificate programs is very limited (CSTA, 2015). Many of those programs that do exist teach very little related to how to teach computer science concepts and instead focus on the computer science skills themselves (Yadev, Gretter, Hambrusch, & Sands, 2016). The result is that teachers at the K-12 level are often teaching computer science with little or no training on computer science education (Yadev, et al., 2016).

Each of the participants in this study had an academic background in some subject other than computer science. This was also found to be common in the existing research. In a 2016 survey by CSTA, over 60% of responding teachers claimed that computer science is not a primary focus of their high school and is instead taught by business or mathematics teachers (Yadev, et al., 2016). In the same survey, 57% of teachers reported that computer science teachers also teach in other non-computer science areas (Yadev, et al., 2016). With a lack of training being compounded by an inability to focus solely on computer science, the teachers often lack the computer science content and pedagogical knowledge required to effectively educate their students (Yadev, et al., 2016).

Given the critical nature of the computer science worker shortage and its potential impact on our economy and national security, the idea that most K-12 computer science teachers appear to come from non-computer science backgrounds is alarming. One of the reasons why computer science professionals do not teach K-12 computer science, as cited by participants in this study, is they can earn far more by working in the computer science industry. Participants in this study also indicated that some administrators work against teachers seeking to develop computer science skills, preventing necessary training out of fear that teachers will leave for the more lucrative computer science industry once trained in the subject. Although outside the scope of this paper, one thing that legislatures and school administrators should contemplate is how to work together to create a compensation system that works for computer science faculty both to attract computer science professionals and to retain those who train into the field.

7.1.1.2 Dearth of classroom-ready resources. The existing research agreed with the findings of this study regarding the need for classroom-ready resources for K-12 computer science teachers. In 2015, Falkner and Vivian conducted a study in which they examined the effectiveness of online resources that were available for K-12 computer science teachers. Citing many of the previously mentioned concerns expressed by computer science teachers, the authors found a dearth of computer science-specific resources for K-12 teachers compared to those found in other subject areas (Falkner & Vivian, 2015). Although online resources that teach coding principles are plentiful, it is difficult and time consuming to find resources focused on computer science pedagogy and curriculum delivery methods (Falkner & Vivian, 2015). One of the biggest challenges for K-12 computer science teachers is determining whether or not a given academic resource is age-appropriate and fit for use in the classroom (Falkner & Vivian, 2015).

Gueudet and Trouche posit that when teachers utilize online resources, they often do not simply introduce them into their classrooms but instead teachers typically study the resources, combine them with other resources, revise them, and present them in their own ways (2015). This methodology of digesting and refining resources creates a sense of ownership of the course content and helps teachers more fully understand the materials themselves (Gueudet & Trouche, 2015). However, the lack of subject-matter expertise among most K-12 computer science faculty forces those teachers to seek out materials which are classroom-ready and can immediately be leveraged with their students (Gueudet & Trouche, 2015).

The lack of subject matter understanding and the dearth of classroom-ready resources are likely connected. If computer science teachers better understood the subject matter, they would likely find existing resources sufficient because they would be able to digest content and morph it for use in their classroom. However, their inability to grasp computer science concepts due to their lack of training in the field creates a need for them to have classroom-ready resources that are immediately usable in the classroom. When those resources are not available, it creates frustration and causes teachers to question their ability to teach effectively, according to participants in this study. The impact negatively affects computer science teacher identity, self-efficacy, and enthusiasm. Administrators and curriculum providers should be aware of the need for classroom-ready resources and the dearth thereof. They should work to ensure existing resources are quality-checked for classroom-readiness and, when a set of resources appears to serve the needs of teachers, those resources should be publicized, adopted, and made available to all teachers in the school or district.

7.1.1.3 Limited availability of collaboration opportunities. Concerns over the lack of collaboration among computer science teachers were also pointed out in the existing research. Computer science teachers at the K-12 level often find themselves as the sole technology teacher at their school which limits their ability to collaborate, share ideas, and learn from their peers (Diethelm, Hildebrandt, & Krekeler, 2009; Yadav, Gretter, & Hambrusch, 2015). The inability to collaborate and share ideas with peers further exacerbates the computer science self-efficacy issues many teachers experience. Many computer science teachers find themselves unpleased with their own class preparation and teaching and, without peers to help them improve, they find themselves struggling to build confidence in their own technology teaching skills (Bender, Schaper, Caspersen, Margaritis, & Hubwieser, 2016).

With limited subject matter knowledge among computer science teachers and with the dearth of classroom-ready resources, the need for collaboration among computer science teachers becomes amplified. If any teacher or group of teachers finds ways to create successful lessons and learning environments for students, it would be in the best interest of teachers, students, and administrators to spread that positivity among the rest of the teaching community. School administrators should be conscious of their computer science teachers and their potential for isolation. They should work with other administrators to encourage and facilitate, perhaps even to require, computer science teacher collaboration and idea sharing.

7.1.2 RQ 2: Impact of the Curriculum Portal

In this study, the curriculum portal was found to have a positive impact on computer science teacher identity, self-efficacy, and enthusiasm. Interview participant Jason, for example, felt that the portal directly improved his identity as a computer science teacher because the layout was so simple and easy to use. He claimed that for the first time he felt that he had all the resources he needed at his fingertips citing, “it gives me a chance to be able to pick and choose what I feel comfortable with, and what I think would benefit the students.”

The portal was also identified by the participants in the study as a tool which positively impacted computer science self-efficacy among teachers. Interview participant MyKayla felt that the layout of the curriculum and portal improved computer science self-efficacy. She posited that many teachers who start out in computer science do not understand basic computer science principles. She explained that if teachers see the lesson modeled in front of them through a well-explained lesson plan, they can visualize it and it can reinforce their confidence that they can be successful in this subject matter. She claimed that the curriculum portal and the way it is laid out enables this type of visualization and builds confidence in teachers using the portal.

The portal was also linked by participants in the study to improved enthusiasm to teach computer science. Interview participant Chase felt that the portal and its easy-to-navigate interface directly impacted his motivation and enthusiasm to teach computer science. He said the portal “is just easy to navigate and easy to get the resources I need to feel more confident. In that sense, the portal helps me with my motivation and identity as a computer science teacher." Fellow interview participant Jason agreed with Chase, citing that he felt lost in the state-approved ECS curriculum but that the organization of the portal where the curriculum and resources are easily navigable and found in the same locations as one another got him more excited about teaching computer science in the ECS I course.

Survey respondents appear to largely agree with the interview participants on the impact of the curriculum portal. When asked if the curriculum portal presents lessons in a way which makes them easier to understand, 17 of the 18 responses indicated in the affirmative, with 10 of the 18 strongly agreeing that it does. See Figure 20 for the full summary of responses to that question.

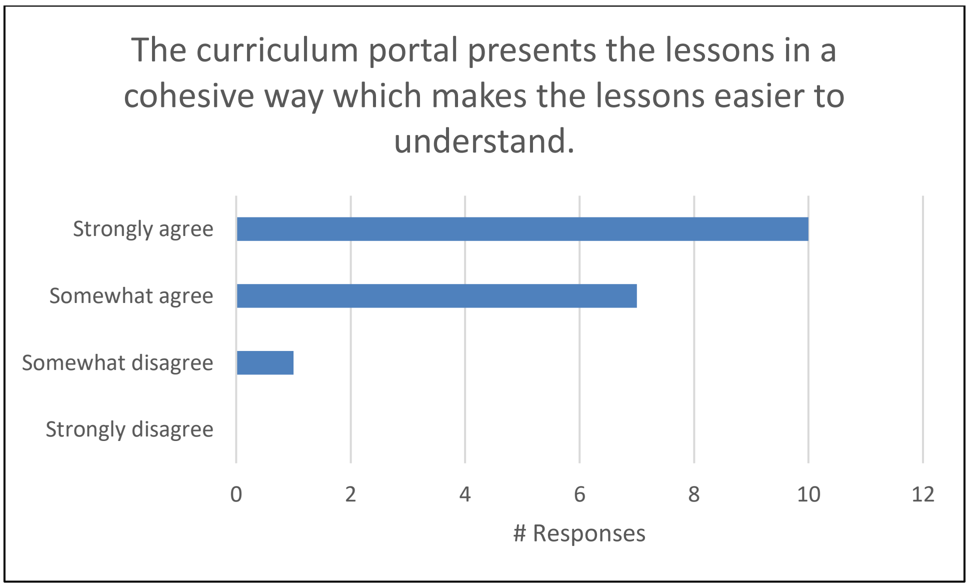


Figure 20 – Does the curriculum portal layout make lessons easier to understand?

The positive impact of the curriculum portal is an interesting finding in this study. Things the participants seemed to find most beneficial regarding the portal are: its simplicity and ease of use; its layout complete with breadcrumbs to return to the previous page easily, the listing of downloadable resources within each course, and the comment section which is found at the bottom of each lesson plan page and is used for sharing ideas between faculty members. These design tips should be considered when designing curriculum resources in the future as they were found in this study to resonate well with computer science teachers.

7.1.3 RQ 3: Impact of the St. Owensby Curriculum

The St. Owensby curriculum enhancements were shown in this study to have positively impacted the computer science teaching identity of the study participants. Georgia felt the St. Owensby curriculum significantly improved her identity as a computer science teacher because of the way the lessons explain topics to teachers. She cited feeling confused reading existing curriculum resources and then obtaining a greater understanding of the lessons and content when reading through the St. Owensby curriculum. Chase also felt the same way, citing that the St. Owensby curriculum builds his computer science teaching identity because “it helps me feel like I can teach the lessons.” He praised the layout of the St. Owensby lessons, saying each topic is laid out in a, “simple lesson plan that I can follow, and I can make my own adjustments, but it has the information that I need to teach the lesson successfully.” He reiterated that such a convenient layout helps develop his sense of computer science teaching identity.

The St. Owensby curriculum was found in this study to have a positive impact on computer science teacher self-efficacy. Chase indicated that he felt frustrated by other curriculum resources because he had to dig through them in order to find what he needed. But with the St. Owensby curriculum enhancements he said he feels that “in 10 minutes I can review anything that I need to brush up on.” He correlates that feeling of empowerment through the curriculum with improved computer science self-efficacy. MyKayla also said the layout of the lesson plans and the way they explain things to teachers helps reinforce her confidence that she can be successful teaching computer science.

The St. Owensby curriculum also had a positive impact on teacher enthusiasm, according to participants in the study. Georgia pointed to the enthusiasm her students have for the St. Owensby lessons and the notion that she loves “anytime a lesson goes great” because she then feels “super-excited to teach it again the next time.” She described some of her students asking for more St. Owensby lessons and telling her they had a great time in her class to the point where they were staying after school to work on the lessons. She cited that when teachers have that kind of success, it significantly impacts their self-efficacy in the subject matter.

The St. Owensby curriculum had several key design factors that, due to the impact the curriculum appears to have had, should be considered when creating curriculum in the future. Those factors are:

1. Designed through collaborative partnership
2. Built by teachers, for teachers
3. Focus on active learning and current, relevant topics
4. Adherence to a central standard

Each of these factors will be discussed throughout the rest of this section.

7.1.3.1 Designed through collaborative partnership. The St. Owensby curriculum enhancements were designed in a partnership of state education leaders, higher education teachers, and K-12 teachers. The different types of individuals comprising this partnership each brought unique expertise to the table. The effectiveness of the partnership was lauded by several of the participants. Curriculum developers and content owners should consider such partnerships when developing future course content. Involvement from various groups including K-12 teachers, higher education organizations, industry, and government can create an atmosphere of collaborative idea-sharing and joint understanding and can result in a more effective course design.

7.1.3.2 Built by teachers, for teachers. Second, the curriculum was built by teachers, for teachers. The participants cited the benefits of having curriculum explained in great detail.

Particularly, participants pointed to the curriculum instructing not just about a lesson, but actually providing advice on how to teach it complete with tips for teachers on sticky points or potential problem areas. Several participants pointed to some of the lessons and how they speak to the teachers specifically about possible areas of concern by using language such as, "you may not understand this concept yet, but don't worry" or "the students may ask about this and if they do, here's what to do." Such language was found to be extremely helpful to the participants in the study as they worked on adding different lessons to their classrooms. Developers of future computer science curriculum should consider using similar notations and explanations to create more classroom-ready resources.

7.1.3.3 Focus on active learning and current, relevant topics. The third key design factor was that the St. Owensby curriculum enhancements focused on active learning and engaging the students through current and relevant topics. Three lessons were pointed out as being particularly engaging by participants in the study: the topics in current technologies lesson (Module 1, Days 1-2); the PacMan lesson (Modeul 1, days 15-16); and the Fortnite lesson (Module 2, Days 4-6). Each of those lesson plans can be found in Appendix F. Interview participants felt these lessons engaged the students partially because the lessons focused on elements of technology that interest the students such as current events and video games. In the topics in current technologies lesson, students investigate current happenings in the world of technology from robotics to space travel to cell phones, mobile, or virtual reality gaming. In the PacMan lesson, the students recreate the PacMan game, which is not a new or modern game but is by most accounts a classic in the gaming world and one with which most students are familiar. In the Fortnite lesson, students dissect aspects of the popular modern video game Fortnite. Participants in the study claimed that when students are engaged in something they know and in which they are interested, it helps reinforce attention and learning. Specifically speaking of the Fortnite lesson, Kristina claimed the lesson worked well because Fortnite is "what they play, it's relevant to them."

When curriculum developers work on future curriculum, care should be taken to ensure content is relevant and interesting to students. Curriculum should be given a consistent and regular refresh to ensure relevancy over long periods of time. Several of the interview participants indicated that the ECS national curriculum was interesting and relevant to students when it was released but now, a decade later, the content has become somewhat stale and in need of updates to make it relevant to today's students. Additionally, some participants pointed out the benefits of getting students to stand up, get away from their desks, and get them interacting with their peers. The PacMan lesson and a few others in the St. Owensby curriculum include group projects or activities that require students to do just that. Interview participant Jason felt that getting the students moving around and interacting added to the effectiveness of the PacMan lesson, "It's a good, get out of your seat, move around activity." Curriculum developers should consider ways to increase interaction of students, particularly if it can get students out of their desks in order to increase attention and interest.

7.1.3.4 Adherence to a central standard. The fourth key factor in the design of the St. Owensby curriculum enhancements was that it adheres to a current central standard. The central standard, in this case, was the existing Utah-approved ECS I curriculum and the state of Utah strands and standards. Administrators at St. Owensby College made the determination to connect to the existing state strands and standards as well as existing ECS resources rather than invent their own standards because they felt it would be more widely adopted and more beneficial to teachers if it fit with what they were currently using in their classrooms. Interview

Participant Georgia indicated that the connection to existing standards is critical. She said the connection to existing ECS resources and the state strands and standards "was so helpful and that's why I used [the St. Owensby curriculum] so often." She continued, however, lamenting that she "would love to see everybody who's trying to help do the same thing." Instead, Georgia said, most external resources create their own format or standard, paying no attention to others, which increases confusion and decreases usability of the resources. Future curriculum and course developers should consider this design factor going forward and do their best to adopt existing standards or integrate with them to increase the utility of their resources.

7.1.4 RQ 4: Impact of Knowledge-Transfer through the Portal

Each of the interview participants claimed the portal had a positive impact on knowledge transfer. Specifically, the comment section located at the bottom of each lesson was called out by multiple participants as being beneficial to them as they learned from other teachers regarding ways they have taught that lesson. Interview participant Chase claimed that the design of the portal enabled him to quickly see other teachers' comments and tips as he read the lesson. He contrasted this to other curriculum resources which include emails or listservs and he pointed out that he would typically delete those because they require him to find the content in question in the email to make use of it and he lacked the time to read through all of those emails. However, the portal, which presents comments alongside the lesson, was extremely beneficial to him mostly because of its convenience. Future resource designers should consider this design element carefully as they create content for K-12 teachers. Recalling some of the difficulties faced by K-12 teachers, particularly the lack of subject matter expertise and the lack of time to focus on computer science, helps us see why resources which are more easily navigable and digestible are preferred over others.

7.1.4.1 Aversion to knowledge transfer. Despite their indications that they had received benefit from other faculty members through the portal comment feature, not a single interview participant had posted their own experience or expertise on the portal. In fact, very few teachers posted comments on the portal even when a chance to win a $100 gift card was provided as incentive. Interview participants offered various reasons why teachers may have not been inclined to share on the portal including: lack of time, lack of confidence or self-efficacy in computer science, or competition and job security. These findings appear to agree with the existing research regarding the lack of subject matter self-confidence and self-efficacy among computer science teachers. Administrators should be conscious of this lack of subject matter expertise and its potential impact on the desire to share and collaborate with other teachers. When designing collaboration opportunities, they should consult with faculty to find ways to help faculty feel safe sharing their thoughts and learning in a collaborative environment.

7.1.4.2 Teachers default to conferences. When asked about ways to improve collaboration among faculty, the interview participants all pointed to the state's summer conference for faculty. Interview participant Jason cited that those attending the conference do so because they want to learn so the conference, in his opinion, would be the best place to facilitate learning about course content. MyKayla felt conferences were an ideal solution to collaboration because they present an environment where teachers can learn from other teachers and where those who might be nervous to present their ideas could do so on a stage with other teachers sharing similar content or thoughts. It was interesting to hear the conference spoken of as a great place to share information. It may show that teachers are more comfortable in a traditional teacher/student setting, which is the format that exists at most conferences, rather than a collaborative peer-to-peer setting. It also may show that teachers are more comfortable in traditional face-to-face training sessions rather than online collaboration opportunities. Even when the geographic difficulties experienced by remote, rural teachers due to their distance from centralized training opportunities were discussed, the teachers felt their rural peers should attend the conference for their training rather than webinars or other technology-related solutions. This may be related to the teachers' general lack of subject matter expertise and, if they truly do not identify as computer science teachers, perhaps they have not fully embraced the benefits of technology solutions for things like professional training. Regardless of the reasons, administrators and training providers should be aware of this proclivity to in-person, teacher/student format for training and learning and build more opportunities to train teachers in formats which are preferable to them.

7.1.5 RQ 5: Impact of Community Partnerships

Every one of the interview participants indicated that community partnerships, such as the one created by St. Owensby administrators, are vital to the success of K-12 computer science education. Brandon Jacobson, from the Utah State Board of Education, called such partnerships a "lifeline to the continuing development of computer science education in Utah and the overall economic growth within Utah's computer science industry" (personal communication, November 19, 2018). Interview participant Jason felt the St. Owensby faculty were instrumental in making the partnership effective because of the different experiences they have in the classroom and how that expertise can help K-12 teachers. He said:

> Listening to [St. Owensby faculty] was great. They lend that adult and professional perspective that [K-12 teachers] might not have. We try and make computer science fun. That's important. But at the same time, [the St. Owensby faculty] lend that professional perspective to the curriculum.

Interview participant Kristina felt partnerships can bring in fresh ideas into the classroom, including critical feedback from industry. She admitted that as teachers at the K-12 level, they "tend not to hear a lot from industry" directly so these partnerships provide a pathway for that input to disseminate to the K-12 teachers. She described the importance of different perspectives as, "Everybody's personalities are slightly different, and you've got to be able to pull from many different brains" in order to build effective curriculum.

Administrators, faculty, industry, and curriculum developers should be cognizant of the impact of collaborative partnerships. More work needs to be done at all levels to encourage the formation of these partnerships and participation in them from teachers, industry professionals, government employees, and administrators.

7.2 Next Steps and Recommendations

Interview participants Jason and Kristina presented ideas for next steps in the development of the St. Owensby curriculum and the collaborative partnership. Jason recommended that St. Owensby create more lessons overall. Specifically, he recommended evaluating the third module in Utah's ECS I curriculum which uses Scratch as an introduction to programming. He felt that those lessons could use another review from a partnership such as the one created for this curriculum and he said he would welcome fresh content from St. Owensby in that module.

Kristina suggested getting more people involved in using the curriculum as a next step. Her recommendation regarding how to do that was to get more involved in the Utah summer education conference by presenting sessions on the lessons and how to implement them.

Another next step that was recommended by members of the partnership was to take the work done on the ECS I course and work to create similar curriculum enhancements to other courses in the state educational system. Some of the courses mentioned in those discussions include: Computer Science Principles, Programming I, and Web Development.

CHAPTER VIII: CONCLUSION

8.1 Implications

The results of this study have implications for positive change at the individual teacher level, the administrator level, and the technology community level. This section will discuss the implications in each of those areas.

At the individual teacher level, the results from this study illustrate many of the challenges facing computer science K-12 teachers. These challenges appear to be formidable and daunting for aspiring K-12 computer science teachers. However, the portal and the St. Owensby curriculum enhancements were found to improve computer science teaching identity, self-efficacy, and motivation among Utah K-12 computer science teachers. The participants involved in the study describe the effects of the portal and curriculum enhancements as instruments that assuaged some of the difficulties faced by themselves and their peers. This study may inspire action to create more partnerships like the one described in this study. As those partnerships create more tools to address the K-12 computer science teacher challenges, teachers will have more assets at their disposal to improve their computer science identity, self-efficacy, and motivation.

For administrators, the challenges cited in this study illustrate a dire situation. As teachers are chosen, assigned, or “voluntold” to teach computer science, they often appear to lack intrinsic computer science teaching identity, self-efficacy, and motivation. This represents a significant challenge from inception: how do you motivate, encourage, reinforce, and inspire somebody to teach a subject in which they are not formally trained and potentially for which they have no inherent passion? The lack of training, tools, and classroom-ready resources available to those teachers further exacerbates these concerns. By embracing and encouraging the development of and adoption of partnerships such as the one outlined in this study, administrators can support the development of more resources to help alleviate some of these teacher challenges. Administrators should focus on improving tools, such as the curriculum portal, to make curriculum more accessible and digestible. Likewise, they should strive to develop computer science curriculum in ways outlined in this study to improve the classroom-readiness of course resources. Lastly, administrators should pay close attention to the collaborative and knowledge-sharing needs of computer science teachers, particularly if their schools are setup in ways which isolate computer science teachers. If administrators work with computer science teachers to identify effective and preferred ways to collaborate and share with their peers, and to remove any concerns about job security in relation to knowledge-sharing, they can dramatically enhance their teachers' ability to improve their understanding of computer science and how to effectively teach the subject.

For the technology community, the computer science worker shortage outlined in this study are real and implications of that shortage are likely felt by most tech employers currently. As described in the study, two core means by which the worker shortage can be addressed in the higher education space are by increasing the interest in computer science among high school students and by increasing retention in computer science at the collegiate level. Within the technology community, industry leaders, practitioners, technology educators and administrators in higher education, and tech- or education-focused legislators should take note of the critical nature of the nation's computer science shortage. These groups should work together to foster the formation of more partnerships such as the one outlined in this study. As more individuals participate in such partnerships, more tools and curriculum can be created to help teachers build passion for computer science within their students.

8.2 Future Research

This study was conducted in the state of Utah with a small sample of high school computer science teachers. In this study, the St. Owensby curriculum and the curriculum portal were found to have a positive impact on the computer science teacher identity, self-efficacy, and enthusiasm of Utah K-12 computer science teachers. An interesting opportunity for future research exists in taking this curriculum and applying it in other states that use the nationwide ECS curriculum and measuring the effectiveness of the curriculum and portal in those states.

Future research into the reasons why teachers are reluctant to share experiences and expertise with others would also be beneficial. Such research could include investigating ways in which teachers are comfortable sharing and/or ways to help teachers overcome their anxiety with sharing their expertise.

Future studies investigating the various ways to improve teacher training would be beneficial. Interview participants defaulted to teacher conferences when asked about the ideal method for training and learning. Investigating ways to help K-12 computer science teachers with learning (and teaching and sharing) via non-traditional means may help them improve their subject matter understanding in a more cost-effective and prudent manner.

This study evaluated the impact of the curriculum portal and the St. Owensby curriculum enhancements on computer science teacher identity, self-efficacy, and motivation. The researcher used interviews and electronic surveys to investigate perceptions of teachers on the impact of the portal and the curriculum enhancements. Another way to evaluate the impact of the portal and the curriculum enhancements would be to implement pre- and post-assessments. A follow up study could be conducted in which the researcher uses a pre-assessment to evaluate a baseline for computer science teacher identity, self-efficacy, and motivation. After the teachers use the portal and the curriculum in a quarter or semester, a post-assessment would then be used to measure any changes. This pre- and post-assessment investigation method can often lead to a more quantifiable measurement of change because it provides benchmarks both before and after an experiment or test.

Another study recommendation which also would leverage pre- and post-assessment is to conduct an evaluation of the computer science teacher identity, self-efficacy, and motivation among various teachers in a pre-assessment and then conduct an experiment in which some of the teachers use their existing ECS curriculum and the others use the St. Owensby curriculum and portal. A post-assessment would then be used to compare the results from both sets of teachers and evaluate which group experienced larger growth in their computer science teacher identity, self-efficacy, and motivation.

Finally, can the type of collaborative partnership and portal and curriculum development in this study be leveraged successfully in other disciplines outside of computer science? Conducting a similar study in a non-computer science discipline would be beneficial to compare the impacts and findings of that study to those presented here.

8.3 Conclusion

There is an underreported crisis happening in the United States of America. This crisis threatens the nation's economic standing in the world, the standard of living of its citizens, and its national security (Adkins, 2012; Napoleoni, 2010; U.S. Department of Education, 2013). The cause of the crisis: a massive, growing gap in the number of qualified computer science workers compared to those needed by the nation's employers (Kalil & Jahanian, 2013; National Center for Education Statistics, 2015). One key factor in the computer science worker shortage is the relatively low numbers of computer science graduates compared to the number of open computer science-related positions. In 2015, there were over 527,000 open computer science jobs in America and there were fewer than 60,000 computer science graduates who entered the workforce (National Center for Education Statistics, 2015).

The computer science dilemma has two core components: student interest and student retention. Although student retention in computer science is a critical piece of the puzzle, it was outside the scope of this study which instead focused on attracting more students to computer science majors. Specifically, this study focused on improving K-12 computer science education to build interest in the discipline among high school students.

The purpose of this study was to evaluate the effectiveness of collaborative online tools, curriculum enhancements, and community partnerships in improving Utah K-12 computer science education. With the massive and growing gap between the number of annual computer science graduates and the number of open computer science jobs, improving interest in computer science as a major for the nation's college students is of paramount importance. By improving the effectiveness of computer science education in the nation's high schools, we may be able to positively impact the number of college students interested in majoring in computer science in the nation's colleges and universities.

This study revolved around measuring the impact of a collaborative partnership of K-12 teachers, higher education, and representatives from the Utah State Board of Education on Utah's K-12 computer science teachers. Through the collaborative partnership, a curriculum portal was designed as well as various curriculum enhancements to supplement existing materials for Utah's ECS I K-12 computer science course. The study utilized mixed methods research to measure the impact of the partnership, the portal, and the curriculum enhancements. Mixed methods research refers to a research methodology that involves both quantitative and qualitative data (Wisdom & Creswell, 2013). By using a convergent design technique, the researcher can acquire both quantitative and qualitative data effectively simultaneously to investigate the research questions from various perspectives (Wisdom & Creswell, 2013).

This study used case study to collect qualitative data through in-person interviews where the researcher investigated participants' worldviews related to the research questions. The case studies in question were Utah-based K-12 classrooms where alternative lesson plans for Utah's ECS I course were implemented.

Various themes emerged throughout the study. One theme was the magnitude of the difficulties facing Utah's K-12 computer science teachers. Many of them do not fully understand basic computer science principles yet they are expected to teach them to their students. Many K-12 computer science teachers have no formal academic training or industry experience in computer science and often they are forced to teach computer science to keep their jobs. There is a general dearth of classroom-ready resources available to K-12 computer science teachers. This forces the teachers to digest materials and modify them for use in their classroom which requires time and expertise that many of them lack. Compounding these issues, K-12 computer science teachers often lack collaboration opportunities because they are the only such teacher in their school.

The curriculum portal was found to have made a positive impact on K-12 computer science teachers. The portal's easy-to-use interface and navigation helped teachers easily find the resources they need to teach their classes. The curriculum enhancements were also found to have made a positive impact on teachers. Participants in the study pointed to several factors in the design of the curriculum which added to its value:

1. The curriculum was designed through collaborative partnership which brought in various perspectives from industry, K-12 teachers, higher education providers, and representatives of the state board of education.
2. The curriculum was built by teachers, for teachers which resulted in the curriculum being articulated clearly with various tips and hints for teachers as they read through the lesson plans.
3. The curriculum focused on active learning and current, relevant topics which resulted in improved engagement from students and increased interest in class materials. Teachers in the study felt that translated to improved learning by the students.
4. The curriculum adhered to a central standard, in this case, Utah's ECS I curriculum and Utah's strands and standards from the state board of education. This created a smooth environment for teachers looking to adopt the curriculum because they did not need to compete against existing expectations regarding curriculum or strands and standards but instead they viewed the curriculum as additive and supporting those standards.

If high school administrators, state representatives, industry professionals, and higher education providers want to work together to make an impact on the growing computer science worker shortage, improving interest in computer science among the nation's high school students is imperative. This study identified design elements of curriculum frameworks and tools as well as curriculum design and format which can be beneficial to those working to improve K-12 computer science education. By leveraging these findings, curriculum resources can be made more easily navigable and digestible by faculty and course content can be made more relevant and interesting to students, creating increased interest in computer science among high school students.

REFERENCES

Adkins, R. C. (2012). *America desperately needs more STEM students: Here's how to get them*. Retrieved from: http://www.forbes.com/sites/forbesleadershipforum/2012/07/09/america-desperately-needs-more-stem-students-heres-how-to-get-them/

Augustine, N. R. (2005). *Rising above the gathering storm: Energizing and employing America for a brighter economic future*. Retrieved August, 29, 2015.

Bailey, J. (2017). *Coding bootcamps have a fundamental problem*. Retrieved from: https://spin.atomicobject.com/2017/10/11/coding-bootcamps-problem/

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Kern, J. (2001). Manifesto for agile software development. Retrieved from: https://moodle2016-17.ua.es/moodle/pluginfile.php/80324/mod_resource/content/2/agile-manifesto.pdf

Bender, E., Schaper, N., Caspersen, M. E., Margaritis, M., & Hubwieser, P. (2016). *Identifying and formulating teachers' beliefs and motivational orientations for computer science teacher education*. Studies in Higher Education, 41(11), 1958-1973.

Bergman, M. M. (2008). *Advances in mixed methods research: Theories and applications*. Sage.

Bobb, K., & Brown, Q. (2017). Access, power, and the framework of a CS education ecosystem. *Moving Students of Color from Consumers to Producers of Technology*. IGI Global.

Bureau of Labor Statistics. (2014). *Employment projections*. Retrieved from: http://www.bls.gov/emp/ep_data_occupational_data.htm

Bureau of Labor Statistics. (2017). *Spotlight on Statistics*. Retrieved from: https://www.bls.gov/spotlight/2017/science-technology-engineering-and-mathematics-stem-occupations-past-present-and-future/home.htm

CSTA. (2015). *National secondary school computer science survey*. Retrieved from http://www.csta.acm.org/Research/sub/Projects/ResearchFiles/CSTA_NATIONAL_SECONDARY_SCHOOL_CS_SURVEY_2015.pdf

Cardelle-Elawar, M., Irwin, L., & Sanz de Acedo-Lizarraga, M. L. (2007). *A Cross Cultural Analysis of Motivational Factors That Influence Teacher Identity*.

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

Chan, W. Y., Lau, S., Nie, Y., Lim, S., & Hogan, D. (2008). *Organizational and personal predictors of teacher commitment: The mediating role of teacher efficacy and identification with school*. American educational research journal, 45(3), 597-630.

Computing Research Association. (2014). *Where the STEM jobs will be*. Retrieved from: http://cra.org/govaffairs/wp-content/uploads/sites/6/2014/10/Newly-Created-STEM-Jobs-Graph-Feb-2014.pdf

Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches* (4th ed.). Thousand Oaks, CA: Sage publications.

Cutts, Q., Robertson, J., Donaldson, P., & O'Donnell, L. (2017). *An evaluation of a professional learning network for computer science teachers*. Computer Science Education, 27(1), 30-53.

Diethelm, I., Hildebrandt, C., & Krekeler, L. (2009). *Implementation of computer science in context – A research perspective regarding teacher-training*. Koli Calling, 97–100.

Duffner, R. (2018). *The rise of the coding boot camp*. Retrieved from: https://www.wired.com/insights/2014/08/rise-coding-boot-camp/

ECS. (2018). *For teachers & districts*. Retrieved from: http://www.exploringcs.org/for-teachers-districts

ECS Utah. (2018). *The Utah exploring computer science initiative*. Retrieved from: http://people.westminstercollege.edu/faculty/hhu/ecs/steps.html

Eccles, J. S., & Wigfield, A. (2002). *Motivational beliefs, values, and goals*. Annual review of psychology, 53(1), 109-132.

Falkner, K., & Vivian, R. (2015). A review of computer science resources for learning and teaching with K-12 computing curricula: An Australian case study. Computer Science Education, 25(4), 390-429.

Fayer, S., Lacey, A., & Watson, A. (2017). *BLS Spotlight on Statistics: STEM Occupations-Past, Present, and Future*.

Feder, M. (2012). *One decade, one million more STEM graduates*. Retrieved from: https://www.whitehouse.gov/blog/2012/12/18/one-decade-one-million-more-stem-graduates

Fisher, A. (2015). *Wanted: Highly skilled tech workers, $100,000-plus salary, no college required.* Retrieved from: http://fortune.com/2015/05/13/devops-jobs/

Forrester. (2018). *Predictions 2018*. Retrieved from: https://go.forrester.com/research/predictions/

Gardner, D. P. (1983). A nation at risk. *Washington, DC: The National Commission on Excellence in Education, US Department of Education*.

Goode, J., Chapman, G., & Margolis, J. (2012). *Beyond curriculum: the exploring computer science program*. ACM Inroads, 3(2), 47-53.

Goode, J., Margolis, J., & Chapman, G. (2014). *Curriculum is not enough: the educational theory and research foundation of the exploring computer science professional development model*. In Proceedings of the 45th ACM technical symposium on Computer science education (pp. 493-498). ACM.

Grasz, J. (2013). *U.S. is producing fewer college graduates with computer and information technology degrees than ten years ago, according to a new study from CareerBuilder and economic modeling specialists*. Retrieved from: http://www.careerbuilder.com/share/aboutus/pressreleasesdetail.aspx?sd=8%2F1%2F2013&id=pr774&ed=12%2F31%2F2013

Gueudet, G., & Trouche, L. (2011). Teachers' work with resources: Documentational geneses and professional geneses. In from text to 'lived' resources (pp. 23-41). Springer, Dordrecht.

Han, X. & Appelbaum, R. (2018). China's science, technology, engineering, and mathematics (STEM) research environment: A snapshot. PLoS ONE 13(4): e0195347.

Harris, L. R., & Brown, G. T. L. (2010). Mixing interview and questionnaire methods: Practical problems in aligning data. *Practical Assessment, Research & Evaluation, 15*(1), 1-19.

Harris, M. M., & Miller, J. R. (2005). Needed: Reincarnation of national defense in education act of 1958. *Journal of Science Education and Technology, 14*(2), 157-172.

Hsieh, H. F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. Qualitative health research, 15(9), 1277-1288.

Holdren, J. P., & Lander, E. (2012). Report to the President—Engage to excel: Producing one million additional college graduates with degrees in science, technology, engineering, and mathematics. *President's Council of Advisors on Science and Technology*.

Hoy, A. W., & Spero, R. B. (2005). *Changes in teacher efficacy during the early years of teaching: A comparison of four measures*. Teaching and teacher education, 21(4), 343-356.

Hu, H. H., Heiner, C., & McCarthy, J. (2016). *Deploying Exploring Computer Science Statewide*. In Proceedings of the 47th ACM Technical Symposium on Computing Science Education (pp. 72-77). ACM.

Joubish, M. F., Khurram, M. A., Ahmed, A., Fatima, S. T., & Haider, K. (2011). Paradigms and characteristics of a good qualitative research. *World Applied Sciences Journal, 12*(11), 2082-2087.

Kalil, T., & Jahanian, F. (2013). *Computer science is for everyone!* Retrieved from: https://obamawhitehouse.archives.gov/blog/2013/12/11/computer-science-everyone

Khan, I. (2016). What is MVC and Why Do we Use MVC? Retrieved from: https://www.c-sharpcorner.com/article/what-is-mvc-and-why-we-use-mvc/

Kunter, M., Frenzel, A., Nagy, G., Baumert, J., & Pekrun, R. (2011). *Teacher enthusiasm: Dimensionality and context specificity*. Contemporary Educational Psychology, 36(4), 289-301.

Launches, P. O. (2009). Educate to Innovate. Campaign for Excellence in Science, Technology, Engineering & Math (Stem) Education.(2009, November 23). *The White House, Office of the Press Secretary. Retrieved from http://www. whitehouse. gov/thepress-*

office/president-obama-launches-educate-innovate-campaign-excellence-sciencetechnology-en.

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. Beverly Hills, CA: Sage Publications, Inc.

Long, J. F., & Hoy, A. W. (2006). *Interested instructors: A composite portrait of individual differences and effectiveness*. Teaching and Teacher Education, 22(3), 303-314.

Mackenzie, N. & Knipe, S. (2006). Research dilemmas: Paradigms, methods and methodology. *Issues in Educational Research, 16*(2). 193-205.

Mahmoud, Q. H. (2005). *Revitalizing computing science education.* Computer, (5), 100-98.

Marra, R. M., & Rodgers, K. A. (2012). Why they're leaving. *JEE Selects, 21*(5). 43.

Menekse, M. (2015). *Computer science teacher professional development in the United States: a review of studies published between 2004 and 2014*. Computer Science Education, 25(4), 325-350.

Merriam, S. B. (1998). *Qualitative research and case study applications in education. Revised and expanded from" case study research in education."*. Jossey-Bass Publishers, 350 Sansome St, San Francisco, CA 94104.

NACE. (2017). *Salary survey: Executive summary*. Retrieved from: http://www.naceweb.org/uploadedfiles/files/2017/publication/executive-summary/2017-nace-salary-survey-winter-executive-summary.pdf

Napoleoni, L. (2010). *Terrorism and the economy: How the war on terror is bankrupting the world*. New York, NY: Seven Stories Press.

National Center for Education Statistics. (2015). *Powerstats*. Retrieved from: http://nces.ed.gov/datalab/powerstats/default.aspx

National Center for Education Statistics. (2016). *Powerstats*. Retrieved from: http://nces.ed.gov/datalab/powerstats/default.aspx

National Science Foundation. (2012). *Science and Engineering Indicators 2012*. Retrieved from: http://www.nsf.gov/statistics/seind12/c0/c0i.htm

National Science Foundation. (2018). *Science and engineering indicators 2018*. Retrieved from: https://www.nsf.gov/statistics/2018/nsb20181/digest

Ni, L., & Guzdial, M. (2012). *Who AM I?: understanding high school computer science teachers' professional identity.* In Proceedings of the 43rd ACM technical symposium on Computer Science Education (pp. 499-504). ACM.

Ni, L., Guzdial, M., Tew, A. E., Morrison, B., & Galanos, R. (2011). *Building a community to support HS CS teachers: the disciplinary commons for computing educators*. In Proceedings of the 42nd ACM technical symposium on Computer science education (pp. 553-558). ACM.

Noonan, R. (2017). STEM Jobs: 2017 Update. *US Department of Commerce Economics and Statistics Administration*.

OECD. (2016). *PISA 2015 Results (Volume I): Excellence and Equity in Education*, PISA, OECD Publishing, Paris, https://doi.org/10.1787/9789264266490-en.

Palys, T. (2008). Purposive sampling. *The Sage Encyclopedia of Qualitative Research Methods.* (2). 697-8.

Patton, M. Q. (1990). *Qualitative evaluation and research methods (2nd ed.).* Newbury Park, CA: Sage Publications.

Rajput, M. (2016). The pros and cons of choosing AngularJS. Retrieved from: https://jaxenter.com/the-pros-and-cons-of-choosing-angularjs-124850.html

Remillard, J. T., & Bryans, M. B. (2004). Teachers' orientations toward mathematics curriculum materials: Implications for teacher learning. Journal for research in mathematics education, 352-388.

Rodriquez, S. (2018). *Jobs are outside Silicon Valley*. Retrieved from: https://www.inc.com/salvador-rodriguez/act-software-developers-map.html

Shaw, E.J., & Barbuti, S. (2010). Patterns of persistence in intended college major with a focus on STEM majors. *The National Academic Advising Association Journal, 30*(2): 19−34.

Smith, M. (2016). Computer science for all. Retrieved from: https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

Soergel, A. (2017). *Trump calls for $200M a year to boost STEM in schools*. Retrieved from: https://www.usnews.com/news/articles/2017-09-25/trump-calls-for-200m-a-year-to-boost-stem-in-schools

The White House. (2011). *Remarks by the president in state of union address*. Retrieved from: https://www.whitehouse.gov/the-press-office/2011/01/25/remarks-president-state-union-address

U. S. Department of Education. (2013). *STEM attrition: College students' paths into and out of STEM fields* (NCES Publication No. NCES 2014-001). Washington, DC: U.S. Government Printing Office.

Watkins, J. & Mazur, E. (2013). Retaining students in science, technology, engineering, and mathematics (STEM) majors. *Journal of College Science Teaching, 42(5)*, 36-41.

Wisdom, J., & Creswell, J. W. (2013). *Mixed methods: integrating quantitative and qualitative data collection and analysis while studying patient-centered medical home models.* Rockville: Agency for Healthcare Research and Quality.

Wiseley, M. (2018). What is ASP.NET and Why Should I Use It? Retrieved from: https://www.wakefly.com/blog/what-is-asp-net-and-why-should-i-use-it/

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). *Expanding computer science education in schools: understanding teacher experiences and challenges*. Computer Science Education, 26(4), 235-254.

APPENDIX A – INTERVIEW TEMPLATE

Demographic Questions:

1. At what school do you teach?
2. How long have you been teaching K-12 overall?
3. How long have you been teaching K-12 computer science?
4. How long have you been teaching ECS I?
5. What grades are students in when they take your ECS I course?
6. What other subjects, if any, do you teach?

Interview Questions:

ECS I Questions:

1. What has been working well for you in terms of teaching ECS I?
2. What have been your primary challenges with ECS I?

Portal Questions:

3. What was your primary interest in using the collaborative curriculum portal?
4. Describe your experience with the portal and its ease of use.
5. How helpful were the collaborative features of the portal?
6. How helpful was the overall layout of the curriculum found in the portal?
7. What changes, if any, would you recommend for the next iteration of the portal?

Curriculum Questions:

8. What was your primary interest in adopting the enhancements to ECS I curriculum?
9. Which alternative lesson plans did you adopt and why?
10. Which alternative lesson plans did you not adopt and why?
11. What were the overall impressions of the curriculum enhancements?
12. Did you notice any differences in terms of student interest or engagement that you attribute to the curriculum enhancements? Why or why not?
13. Going through each of the adopted lesson plans: what did you like about the lesson plan and what did you not like?
14. What other enhancements to ECS I would you recommend?

Partnerships Questions:

15. What are your overall feelings regarding the impact of community partnerships like the one created for ECS I curriculum?
16. How can we get more people, companies, and institutions to join or start similar community partnerships?

RQ Questions:

17. Do you consider yourself a computer science teacher? How has that changed since using the portal and curriculum enhancements?
18. Describe your confidence as a computer science teacher. How has that changed since using the portal and curriculum enhancements?
19. Describe your enthusiasm to teach computer science. How has that changed since using the portal and curriculum enhancements?

APPENDIX B – CURRICULUM AND PORTAL SURVEY

COMPUTER SCIENCE AND SOFTWARE ENGINEERING

AUBURN UNIVERSITY

SAMUEL GINN COLLEGE OF ENGINEERING

(NOTE: DO NOT AGREE TO PARTICIPATE UNLESS IRB APPROVAL INFORMATION WITH CURRENT DATES HAS BEEN APPLIED TO THIS DOCUMENT.)

INFORMATION LETTER
for a Research Study entitled
"The Effects of Community Partnership on K12 Computer Science Education "

You are invited to participate in a research study to evaluate the effectiveness of the curriculum portal and the curriculum enhancement content. The study is being conducted by Aaron Reed, Ed.D., under the direction of Dr. Jakita O. Thomas in the Auburn University Department of Computer Science and Software Engineering. You are invited to participate because you are a registered user of the curriculum portal and are age 19 or older.

What will be involved if you participate? If you decide to participate in this research study, you will be asked to complete a short electronic survey. Your total time commitment will be approximately 5-10 minutes. In the survey, you will be asked if you would be willing to provide additional information in an interview setting. If you wish to do so and are selected, you will be asked to participate in a one-on-one interview with Dr. Reed that is expected to last 30-60 minutes.

Are there any risks or discomforts? The risks associated with participating in this study are limited to a potential breach of confidentiality by the researcher. To minimize these risks, we will ensure that all data is confidential and that your identifying information is never combined with the data you provide.

There is no cost associated with your participation and there is not compensation awarded for completing the survey.

If you change your mind about participating, you can withdraw at any time during the study. Your participation is completely voluntary. If you choose to withdraw, your data can be withdrawn as long as it is identifiable. Your decision about whether or not to participate or to stop participating will not jeopardize your future relations with Auburn University, the Department of Computer Science and Software Engineering or Dr. Reed.

Any data obtained in connection with this study will remain confidential. We will protect your privacy and the data you provide by keeping it secure and removing any identifying information. Information collected through your participation may be useful in identifying new ways to improve the curriculum portal and its contents.

If you have questions about this study, please ask them now or contact Aaron Reed at arr0053@tigermail.auburn.edu.

If you have questions about your rights as a research participant, you may contact the Auburn University Office of Research Compliance or the Institutional Review Board by phone (334)-844-5966 or e-mail at IRBadmin@auburn.edu or IRBChair@auburn.edu.

HAVING READ THE INFORMATION PROVIDED, YOU MUST DECIDE IF YOU WANT TO PARTICIPATE IN THIS RESEARCH PROJECT. IF YOU DECIDE TO PARTICIPATE, THE DATA YOU PROVIDE WILL SERVE AS YOUR AGREEMENT TO DO SO. YOU ARE WELCOME TO PRINT THIS LETTER AND KEEP A COPY OF THIS STUDY INFORMATION LETTER AND CONSENT FORM FOR YOUR RECORDS.

The Auburn University Institutional Review Board has approved this Document for use from
11/01/2018 to 10/31/2019
Protocol # 18-398 EP 1810

1. What is your role in relation to the ECS I curriculum?

- ◯ Teacher
- ◯ Administrator
- ◯ Curriculum provider/editor
- ◯ Other

2. How long have you been using the ECS I course?

- ◯ Over 6 months
- ◯ 4-6 months
- ◯ 2-4 months
- ◯ Less than 2 months

3. At what level have you been teaching the ECS I course?

- ◯ 9th grade
- ◯ 10th grade
- ◯ 11th grade
- ◯ 12th grade

Other (please specify)

4. Of the 20 available alternative lesson plans for ECS I, how many did you implement?

- ◯ 0
- ◯ 1-4
- ◯ 5-8
- ◯ 9-12
- ◯ 13 or more

5. The curriculum portal was an extremely valuable tool in helping me become an effective ECS I educator.

- ◯ Strongly agree
- ◯ Somewhat agree
- ◯ Somewhat disagree
- ◯ Strongly disagree

(Optional) Comments:

6. The curriculum portal presents the lessons in a cohesive way which makes the lessons easier to understand.

- ◯ Strongly agree
- ◯ Somewhat agree
- ◯ Somewhat disagree
- ◯ Strongly disagree

(Optional) Comments:

7. After reading through a given lesson on the curriculum portal, I have a better understanding of how to effectively teach the lesson.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree
- Comment (optional)

(Optional) Comments:

8. After reading through a given lesson on the curriculum portal, I have a better understanding of the desired lesson outcomes.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

(Optional) Comments:

9. After reading through a given lesson on the curriculum portal, I feel more confident to deal with student questions on the lesson topic.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

(Optional) Comments:

10. After reading through the resources in the curriculum portal, I feel more comfortable using web-based resources for teachers.

- ◯ Strongly agree
- ◯ Somewhat agree
- ◯ Somewhat disagree
- ◯ Strongly disagree

(Optional) Comments:

11. After reading through the resources in the curriculum portal, I feel more confident in my ability to engage the students in the class.

- ◯ Strongly agree
- ◯ Somewhat agree
- ◯ Somewhat disagree
- ◯ Strongly disagree

(Optional) Comments:

12. By having access to both the Utah ECS I curriculum and the resources in the curriculum portal, I feel I will be more prepared to teach the ECS I course in the future.

- ◯ Strongly agree
- ◯ Somewhat agree
- ◯ Somewhat disagree
- ◯ Strongly disagree

(Optional) Comments:

13. The combination of the Utah ECS I curriculum and the resources in the curriculum portal makes me more enthusiastic about teaching the ECS I course in the future.

- ◯ Strongly agree
- ◯ Somewhat agree
- ◯ Somewhat disagree
- ◯ Strongly disagree

(Optional) Comments:

14. The combination of the Utah ECS I curriculum and the resources in the curriculum portal makes me more enthusiastic about teaching the ECS I course in the future.

- ◯ Strongly agree
- ◯ Somewhat agree
- ◯ Somewhat disagree
- ◯ Strongly disagree

(Optional) Comments:

15. The additional resources in the curriculum portal helped me build student interest in computer science in my class.

- ◯ Strongly agree
- ◯ Somewhat agree
- ◯ Somewhat disagree
- ◯ Strongly disagree

(Optional) Comments:

16. The additional resources in the curriculum portal helped me more effectively engage the students in my class.

- ◯ Strongly agree
- ◯ Somewhat agree
- ◯ Somewhat disagree
- ◯ Strongly disagree

(Optional) Comments:

17. Would you recommend this curriculum to other teachers (why or why not)?

○ Yes ○ No

(Optional) Comments:

18. Would you be willing to participate further in an interview setting? Note: participation is completely voluntary. Interviews will last 30-60 minutes and will be conducted via phone, video conference, or in-person (your choice). All data collected in interviews will be strictly confidential and any published results from the study will be anonymous. Those who are willing to participate in interviews and are selected for interviews will receive a $25 Amazon gift card to thank you for your time.

If you are willing to participate in an interview, enter your name, email, and phone number below. Leaving the fields below blank indicates you are not willing to participate in an interview.

Name

Email

Phone

Thank you for your time!

OK

APPENDIX C – EMAIL INVITATION FOR SURVEY

Dear ______________________,

I am a PhD graduate student in the Department of Computer Science and Software Engineering at Auburn University. I would like to invite you to participate in my research study to evaluate and improve the curriculum portal software and the curriculum enhancement content contained therein. You may participate if you are familiar with the curriculum portal software and/or implemented some of the curriculum enhancements contained within the portal.

If you agree to participate, you will be asked to fill out a short electronic survey (should take 5-10 minutes). You can find the survey here: (URL to be included)

All survey data will be completely confidential to the public. I am the only person who will have your identifying information. I keep that information only because, if there are questions about your response(s) and you agree in the survey to let me ask you some follow up questions about your responses, I may contact you to discuss further. In such a case, all your information will be completely confidential. The goal of this study is to provide the curriculum portal team and curriculum team with more information regarding how their system and content are used and how it could be improved. Participation in this study is completely voluntary and you are under no obligation to participate.

If you would like to know more information about this study, an information letter is attached to this email. If you decide to participate after reading the letter, you can access the survey from the link provided above.

If you have any questions, please contact me at arr0053@tigermail.auburn.edu or my advisor, Dr. Jakita O. Thomas at jnt0020@auburn.edu.

Thank you for your consideration,

Aaron Reed

APPENDIX D – INFORMATION LETTER AND CONSENT FORM

COMPUTER SCIENCE
AND SOFTWARE
ENGINEERING

AUBURN UNIVERSITY

SAMUEL GINN
COLLEGE OF ENGINEERING

(NOTE: DO NOT SIGN THIS DOCUMENT UNLESS AN IRB APPROVAL STAMP WITH CURRENT DATES HAS BEEN APPLIED TO THIS DOCUMENT.)

INFORMATION LETTER
for a Research Study entitled
"The Effects of Community Partnership on K12 Computer Science Education"

You are invited to participate in a research study to evaluate the effectiveness of the curriculum portal and the curriculum enhancement content. The study is being conducted by Aaron Reed, Ed.D., under the direction of Dr. Jakita O. Thomas in the Auburn University Department of Computer Science and Software Engineering. You are invited to participate because you are a registered user of the curriculum portal and are age 19 or older.

What will be involved if you participate? If you decide to participate in this research study, you will be asked to complete a short electronic survey. Your total time commitment will be approximately 5-10 minutes. In the survey, you will be asked if you would be willing to provide additional information in an interview setting. If you wish to do so and are selected, you will be asked to participate in a one-on-one interview with Dr. Reed that is expected to last 30-60 minutes.

Are there any benefits to participating? The research will be used to further K-12 computer science education. Participants can take pride in playing a role in furthering that field of research. After the survey responses have been collected, a drawing will be conducted in which one survey respondent will be selected at random to receive a $50 gift card to Amazon.com. Participants who complete the survey will have a 1/(number of survey responses submitted) chance to win that gift card. All interview participants will receive a $25 gift card to Amazon.com as a way of thanking you for your time. **Are there any risks or discomforts?** The risks associated with participating in this study are limited to a potential breach of confidentiality by the researcher. To minimize these risks, we will ensure that all data is confidential and that your identifying information will be removed from the data you provide.

There is no cost associated with your participation.

Participant's Initials: _______ Page 1 of 2

3101 Shelby Center, Auburn, AL 36849-5347; Telephone: 334-844-4330; Fax: 334-844-6329
www.auburn.edu

COMPUTER SCIENCE
AND SOFTWARE
ENGINEERING

AUBURN UNIVERSITY

SAMUEL GINN
COLLEGE OF ENGINEERING

If you change your mind about participating, you can withdraw at any time during the study. Your participation is completely voluntary. If you choose to withdraw, your data can be withdrawn as long as it is identifiable. Your decision about whether or not to participate or to stop participating will not jeopardize your future relations with Auburn University, the Department of Computer Science and Software Engineering or Dr. Reed. You may opt out of answering any question and you may close your browser before completing the survey.

Any data obtained in connection with this study will remain confidential. We will protect your privacy and the data you provide by keeping it secure and removing any identifying information. Information collected through your participation may be useful in identifying new ways to improve the curriculum portal and its contents.

If you have questions about this study, please contact Aaron Reed at arr0053@tigermail.auburn.edu.

If you have questions about your rights as a research participant, you may contact the Auburn University Office of Research Compliance or the Institutional Review Board by phone (334)-844-5966 or e-mail at IRBadmin@auburn.edu or IRBChair@auburn.edu.

HAVING READ THE INFORMATION PROVIDED, YOU MUST DECIDE IF YOU WANT TO PARTICIPATE IN THIS RESEARCH PROJECT. IF YOU DECIDE TO PARTICIPATE, THE DATA YOU PROVIDE WILL SERVE AS YOUR AGREEMENT TO DO SO. THIS LETTER IS YOURS TO KEEP.

____

Investigator's signature Date

____

Print Name

3101 Shelby Center, Auburn, AL 36849-5347; Telephone: 334-844-4330; Fax: 334-844-6329

www.auburn.edu

COMPUTER SCIENCE
AND SOFTWARE
ENGINEERING

AUBURN UNIVERSITY

SAMUEL GINN
COLLEGE OF ENGINEERING

(NOTE: DO NOT SIGN THIS DOCUMENT UNLESS AN IRB APPROVAL STAMP WITH CURRENT DATES HAS BEEN APPLIED TO THIS DOCUMENT.)

INFORMED CONSENT
for a Research Study entitled
"The Effects of Community Partnership on K12 Computer Science Education"

You are invited to participate in a research study to evaluate the effectiveness of the curriculum portal and the curriculum enhancement content. The study is being conducted by Aaron Reed, Ed.D., under the direction of Dr. Jakita O. Thomas in the Auburn University Department of Computer Science and Software Engineering. You are invited to participate because you are a registered user of the curriculum portal and are age 19 or older.

What will be involved if you participate? If you decide to participate in this research study, you will be asked to participate in a one-on-one interview with Dr. Reed that is expected to last 30-60 minutes.

Are there any benefits to participating? The research will be used to further K-12 computer science education. Participants can take pride in playing a role in furthering that field of research. All interview participants will receive a $25 gift card to Amazon.com as a way of thanking you for your time.

Are there any risks or discomforts? The risks associated with participating in this study are limited to a potential breach of confidentiality by the researcher. To minimize these risks, we will ensure that all data is confidential and that your identifying information will be removed from the data you provide. Your responses will be used in the study with a pseudonym to protect your confidentiality.

There is no cost associated with your participation.

Participant's Initials: _____

3101 Shelby Center, Auburn, AL 36849-5347; Telephone: 334-844-4330; Fax: 334-844-6329
www.auburn.edu

COMPUTER SCIENCE
AND SOFTWARE
ENGINEERING

AUBURN UNIVERSITY

SAMUEL GINN
COLLEGE OF ENGINEERING

If you change your mind about participating, you can withdraw at any time during the study. Your participation is completely voluntary. If you choose to withdraw, your data can be withdrawn as long as it is identifiable. Your decision about whether or not to participate or to stop participating will not jeopardize your future relations with Auburn University, the Department of Computer Science and Software Engineering or Dr. Reed. To withdraw from the study, simply email Aaron Reed at arr0053@tigermail.auburn.edu expressing a desire to withdraw and your responses will be deleted.

Any data obtained in connection with this study will remain confidential. We will protect your privacy and the data you provide by keeping it secure and removing any identifying information. Information collected through your participation may be useful in identifying new ways to improve the curriculum portal and its contents.

If you have questions about this study, please ask them now or contact Aaron Reed at arr0053@tigermail.auburn.edu.

If you have questions about your rights as a research participant, you may contact the Auburn University Office of Research Compliance or the Institutional Review Board by phone (334)-844-5966 or e-mail at IRBadmin@auburn.edu or IRBChair@auburn.edu.

HAVING READ THE INFORMATION PROVIDED, YOU MUST DECIDE WHETHER OR NOT YOU WISH TO PARTICIPATE IN THIS RESEARCH STUDY. YOUR SIGNATURE INDICATES YOUR WILLINGNESS TO PARTICIPATE.

Participant's signature	Date	Investigator's signature	Date
Printed name		Printed name	

Page 2 of 2

3101 Shelby Center, Auburn, AL 36849-5347; Telephone: 334-844-4330; Fax: 334-844-6329

w w w . a u b u r n . e d u

APPENDIX E – EMAIL INVITATION FOR INTERVIEW

Dear ______________________,

I'd like to thank you for your willingness to complete the electronic curriculum portal survey. On that survey, you indicated that you would not be opposed to participating in an interview to discuss your thoughts on the curriculum and the portal more in depth. I would like to invite you to participate in an interview to evaluate and improve the curriculum portal software and the curriculum enhancement content contained therein. The goal of this study is to provide the curriculum portal team and curriculum team with more information regarding how their system and content are used and how it could be improved.

If you agree to participate, I will contact you to setup a date/time for the interview (should take 30-60 minutes). Interviews can take place on the phone, on video conference call, or in person at your office or mine (whichever you prefer).

Interviews will be recorded and transcribed for use in the study. All identifying information will be removed from data outside of the interview recording. Data will be kept confidential and all published findings will be anonymous. Participation in this study is completely voluntary and you are under no obligation to participate.

In case you would like to know more information about this study, an informed consent form is attached to this email which contains more details about the study and what it means to participate. You can also contact me directly if you want any clarifying details or have any questions. If you'd like to participate, simply reply to this email indicating your desire to participate in an interview.

If you have any questions, please contact me at arr0053@tigermail.auburn.edu or my advisor, Dr. Jakita O. Thomas at jnt0020@auburn.edu.

Thank you for your consideration,

Aaron Reed

APPENDIX F – ST. OWENSBY CURRICULUM ENHANCEMENTS

Module 1: Days 1-2

Instructional Days: 1-2 (Topic description, objectives, and ECS outline can be found on pages 30-31 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

Outline of the Lessons:

- Computer science careers (10 minutes)
- CS current events (30 minutes)
- Finding new tech (30-60 minutes)
- Industry/Company evaluation (40-70 minutes)

Student Activities:

- Discuss computer science careers
- Discuss computer science current events
- In groups, find new technology that others in the class may have never heard of
- Present new technology as a group
- In groups, identify an industry or company and evaluate its usage of computers
- Present industry/company as a group

Teaching/Learning Strategies:

- Computer science careers (10 minutes)
    - The goal here is to provide the students with a basic understanding of how many jobs there are in computer science, how well they pay, and what types of industries in which they can work in CS.
        - There are more jobs than Americans can fill – its' one of the fastest growing fields in America (https://www.bls.gov/spotlight/2017/science-technology-engineering-and-mathematics-stem-occupations-past-present-and-future/pdf/science-technology-engineering-and-mathematics-stem-occupations-past-present-and-future.pdf)

        - Jobs pay incredibly well (~$65k right out of college)
        - They can work in any industry! Sports, medicine, military, food, etc… anything!
- CS current events (30 minutes)
    - Show 2-3 videos highlighting ways in which technology is changing our world. Ideally these are concepts that would be extremely interesting or relevant to the students and get their creative juices flowing and build their passion for tech. After each video, hold a discussion with the students regarding your passion for that technology, how it excites them, the ramifications of that technology, where this might be going in the future, and so on. The goal is simply to show interesting, cool, and innovative technology and get the students excited about tech. Here are some examples:
        - Mars One: https://www.youtube.com/watch?v=n4tgkyUBkbY
        - Farming robots: https://www.youtube.com/watch?v=ENf2hgWDiM8
        - Dog robot: https://www.youtube.com/watch?v=NtU9p1VYtcQ
        - Hololens: https://www.youtube.com/watch?v=pLd9WPlaMpY&t=65s
- Introduce and define terms: computer and computing (5 minutes)
- Finding new tech (30-60 minutes)
    - Have the students divide into groups. The goal for each group is to scour the web and find some new technology that they think perhaps nobody in the class has ever heard of. Teach the students how to do so in a quick demo by pulling up a search engine such as Google and typing a search term. Each team should find one video or article describing a technology that is new and interesting to them. Have the groups present their findings to the class.
- Industry/Company evaluation (40-70 minutes)
    - In the same groups, students are to identify an industry (sports, medicine, food service, etc) or a company (Apple, McDonald's, Disney, etc). In that industry/company, they are to brainstorm as a group:
        - How are computers used in this industry/company?
        - What problems are they solving with technology?
        - Could they solve these problems without technology?
        - What other problems do they see in this industry/company?
        - How do they see the future of this industry/company progressing, particularly related to its use of computers?

Module 1: Days 3-4

Instructional Days: 3-4 (Topic description, objectives, and ECS outline can be found on pages 32-35 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and

lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

Outline of the Lessons:

- Computer function activity (20 minutes)
- Computer simulation activity (40 minutes)
- Computer simulation competition (20 minutes)

Student Activities:

- Participate in a human computer exercise
- Compete with classmates in a human computer exercise

Teaching/Learning Strategies:

- Computer function activity (20 minutes)
    - The goal of this activity is to help students understand *computing* and the concept that computers are exceptionally good at computing – far better than humans. This is an individual activity in which you are to provide the students with a series of computational steps to repeat in sequence, demanding 100% accuracy and the fastest possible speed. Feel free to use different types of computational steps and inputs for this exercise but here is an example:
        - Provide the students with a list of phrases written on a piece of paper or the whiteboard. An example set of phrases is:
            - "How are you today?"
            - "I like pizza."
            - "I can't wait for the weekend."
            - "Why is the sky blue?"
            - "What's your name?"
            - "I remember when I was just a kid."
            - "I love this class!"
            - "Who are you?"
            - "My friend, it's good to see you again."
            - "Let's talk again at 3 pm today"
        - Then you'll have the students pull out a piece of paper and instruct them that for each phrase, they are to perform certain computations and write down the results on their paper. For example:
            - How many characters are in the phrase?
            - How many vowels are in the phrase?
            - How many punctuation marks are in the phrase?
            - How many numbers are in the phrase?
            - What is the phrase written in reverse?
            - What is the phrase written after removing all vowels and the letters "t", "l", and "s"?
            - What is the phrase written after sorting the words alphabetically?

        - Start a timer and give the students 10-15 minutes to get as far through the phrases as they can. (Optional) Give a candy bar or some kind of reward for the winner. Before declaring a winner, have the students pass their sheet to a neighbor and have it graded – 100% accuracy is required in order to win.
        - Have a brief discussion with the students about the activity they just did. Did they enjoy it? Why or why not? Would they like to do work like that for the rest of their lives as a career? Why or why not? Discuss with students the principle that computers function quite differently from humans. If given a set of simple instructions (e.g., sort this list of names, calculate the product of these numbers, etc), the computer will be able to execute those instructions flawlessly, incredibly quickly, and forever, as long as it has power. The computer will never get bored and never slow down. Humans simply cannot function that way. In many respects, this is why computers are so prevalent in the world – they can do these sorts of things and humans cannot.
        - Continue the discussion by asking a question: “Is there anything humans do that computers cannot?” There are a ton of examples and mostly they relate to creativity, sociability, thinking autonomously. For example, a person can very easily answer questions such as “How are you feeling today?”, “What’s your favorite color?”, and “Do you prefer plain or peanut M&M’s?” whereas a computer has a very difficult time being creative or thinking abstractly in those ways.
- Computer simulation activity (60 minutes)
    - In this activity, students will be mimicking the roles of various computer components to better understand the definitions, purposes, and roles of those components within a computer system. This is meant to be an introductory activity to give students a very basic understanding of the computer components, not an in-dept discussion on the architecture of computers.
    - An important thing to help students understand is that all components of a computer have an input, an output, and a computing task. The computing task dictates to that piece of hardware what it should do with the input and how it should handle output. The computer does not think for itself, nor do the computer’s components. They are very literal entities and will perform their computing task exactly and precisely.
    - Components will include:
        - Input device (keyboard/mouse)
            - Input (gets data from a user)
            - Output (sends data to the volatile storage)
            - Computing task (none) – merely passes data along
        - Volatile storage (RAM)
            - Input: gets data from input device (or programs)
            - Output: sends data to the CPU or persistent memory when requested
            - Computing task: organization of storage of items in memory.
        - Processor (CPU)
            - Input: gets data from the volatile storage

            - Output: sends data to persistent storage
            - Computing task: performs calculations such as sorting, arithmetic, etc.
        - Persistent storage (e.g., hard drive, flash drive, CD/DVD, etc)
            - Input: gets data from CPU or volatile memory
            - Output: sends data to the output devices when requested
            - Computing task: organization of storage of items in memory.
        - Visual output (monitor)
            - Input: gets data from persistent storage
            - Output: sends data to the user in visual form
            - Computing task: translate data to whatever visual form is required
        - Audio output (sound card and speakers)
            - Input: gets data from persistent storage
            - Output: sends data to the user in audio form
            - Computing task: translate data to whatever audio form is required
    - Assign students in the class to each of the various components. You will then pass data to the input device students and they will perform the calculation. Here is an example of a program:
        - Program purpose: calculate 5*3 and if the answer is 15, show "Correct" and make a bell sound. If the answer is not 15, show "Wrong!" play a sound of somebody crying.
            - Input: (in order)
                - 5
                - *
                - 3
                - 15
                - Bell sound
                - Crying sound
        - Pass input to the input team either by telling them the input "saying 5, times, 3, etc" or by writing the inputs on a card and passing them individually to the team. The input team passes the input to the volatile memory team. The volatile memory team is to store the memory. This can be simulated by drawing some memory slots on a white board and they fill in those slots or by using post-it notes on a wall, etc. Once memory is stored, you have the volatile memory team pass it to the CPU. The CPU performs the multiplication and passes the result to the persistent memory. The answer (in the case above, 15) is passed to persistent memory from volatile memory as well along with the two sound inputs. The persistent memory sends a signal to the output (the words "correct/wrong" to the visual output and the corresponding audio sound to the audio output). The output teams output their responses and the program is finished.
        - You can then have several discussions with the class:
            - You can discuss the fact that the CPU is the "brain" of the computer. It does all the processing and everything else is passing data around. This

is an oversimplification for modern computers but it works for this example. Ask the students what processing is happening outside of the CPU in this model. Two possible answers are that you need to compare the result of the 5*3 operation to 15 and see if they are the same. That happened in persistent memory in our example but should happen in the CPU. Likewise, the decision to pass correct/wrong to visual output and the corresponding sound to audio output is a comparison or CPU decision and it was also happening in the persistent memory. Get the students to help you refine the computer so that the processing all happens in the CPU. Come up with some revised rules for the computer based on these discussions and repeat the exercise.

- You can play with memory sizes to limit the number of spaces you have available in volatile and persistent memory. Show the students that if you don't have enough memory, you can't perform the operations. Explain that this is a very simple example, but that in real computers, RAM and hard drive size do dictate what you can do on a computer (insufficient RAM will limit your ability to run the most modern programs and games and insufficient hard disk space will limit how much you can store on your computer).
- You can play with power. Simulate a power outage when the memory is still only in the volatile memory and not in persistent memory. When the power goes out, you wipe the board and the program is lost. Use this to explain the difference between volatile and persistent memory. Why isn't all memory persistent? It stores data through power outages but it is slower than volatile memory. Talk about types of persistent storage outside of hard disks including flash drives, CD/DVDs, etc.
- Talk about the overall rules for the computer. Who dictates what goes where and when? Explain that it is a program and operating system that make that happen in computers.

- Computer simulation competition (30 minutes)
    - Once the class exercise is complete, continue the exercise with different programs but divide the students into teams or multiple computers to see who can complete the programs faster but with 100% accuracy.

Module 1: Days 5-7

Instructional Days: 5-7 (Topic description, objectives, and ECS outline can be found on pages 36-41 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/**.**

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to

teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

Outline of the Lessons:

- Search skills (20 minutes)
- Researching and evaluating information (30 minutes)
- Distinguishing good data from bad (30 minutes)
- Explore advanced search techniques (30 minutes)
- Activity – KSL.com's "Spot the Fake News" weekly quiz (30 minutes)
- Establishing confidence in data (30 minutes)

Student Activities:

- Participate in group discussions
- Participate in hands-on search engine exercises
- Take fake news quiz, think about the ramifications of fake news
- Identify ways to establish confidence in data and share thoughts/experiences with peers

Teaching/Learning Strategies:

- Search skills (20 minutes)
    - Run a few simple searches
        - What is the most used/downloaded social media app?
        - Find an amazing, but simple life hack tutorial.
        - Which celebrity search is responsible for the most online viruses?
    - Point out/help identify search results that are actually paid ads
    - Demonstrate the virtue and value of looking at the second or third page of results for a well-rounded understanding of the data
- Researching and evaluating information (30 minutes)
    - Go to dhmo.org, an informative site about the dangers of dihydrogen monoxide
        - Yes, this site is about water (H_2O), but most students won't recognize that. DO NOT TELL THEM YET!
    - Have the students read through the site and answer the following questions:
        - What is the problem according to this site?
        - What do you think we should do about it?
    - Allow the students to give ultra-brief explanations and proposals
    - Reveal the truth about DHMO.org
    - Discuss with the students the concept that there is bad info on the internet masquerading as valid truth
    - Show State Farm commercial involving the "French male model" to help students draw a parallel
        - https://www.youtube.com/watch?v=3DZbSlkFoSU
- Distinguishing good data from bad (30 minutes)

- Discuss bad data
  - Review the experience from yesterday with DHMO.org
    - Should take no more than 10 minutes
- Identifying bad search results
  - Prepare some safe-for-school searches that result in sketchy, not-entirely-true data results
  - Where possible, compare good and bad results briefly
  - As a class, create a list of criteria that indicate a result is "bad" (not trustworthy)
  - As a class, create a list of criteria that indicate a result is good

- Explore advanced search techniques (30 minutes)
  - Using quotations for exact value search results
  - Search social media only by using @ in front of the search value
  - Search for a specific price by using $ in front of a number
  - Exclude words from your search results using – in front of the words to be excluded
    - For more advanced techniques, check out Google's cheat sheet
    - https://support.google.com/websearch/answer/2466433?hl=en
  - Provide the students with some things to search on and experiment with the advanced searching techniques.
- Activity – KSL.com's "Spot the Fake News" weekly quiz (30 minutes)
  - A fun quiz that is updated weekly
  - You can let students research each headline to verify the information before answering each question
  - https://www.ksl.com/?sid=46257703&nid=148&title=quiz-can-you-spot-the-fake-news-stories
  - Discuss the stories in the fake news quiz, the students' thoughts and responses, and how to spot fake news.
- Establishing confidence in data (30 minutes)
  - Give students the data rubric
  - Use the rubric as currently established in the ECS curriculum
  - Work as a class to create a better, more refined rubric for verifying the veracity of search result data
  - Practice the rubric in the following ways
    - Find articles that report the same incident or event in different, possibly biasing ways
      - Example: Search "California school teacher shoots gun in class" and check out the headlines
    - "Student recovering after teacher accidentally fires gun in NorCal classroom"
    - "Teacher accidentally fires gun, injures student"
    - "California teacher accidentally discharges weapon in school during class, police say; students injured"
    - "Gun-Trained Teacher Accidentally Shoots Gun In Calif. High School Classroom"

- Have a discussion about what might be true (similarities between articles), what might not be true (inflammatory or dismissive statements), and what cannot be confirmed (speculations, interpretations, and discrepancies)
- Talk about data biasing opinion (how the data makes the reader "feel")
- Introduce a safe-for-school article from The Onion News (theonion.com)
  - The Onion is a satire site that many people confuse for real news
    - Discuss why that is

Module 1: Days 8-9

Instructional Days: 8-9 (Topic description, objectives, and ECS outline can be found on pages 42-45 of the ECS 8 curriculum)
Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.
This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.
Outline of the Lessons:

- Pick some combination of the following to create two days of content:
  - Discussion of computers in communication (30 minutes)
  - Discussion of progression of communication technology and what's next (30 minutes)
  - Email your peers exercise (60 minutes)
  - Digital Footprint discussion (60 minutes)

Student Activities:

- TODO

Teaching/Learning Strategies:

- Discussion of computers in communication (30 minutes)
  - The goal of this exercise is to get the students to critically think about the progression of technology in communications past, present, and future.
  - Start with asking the students to list, either individually, in groups, or on the whiteboard as a class, all the ways in which people communicate. The only wrong answer is something that is not a means of communication. The list will likely involve everything from talking in person and handshakes to texting and Snapchat.
  - Now have the students cross off the list all means of communication which require computers. Note that even landline telephones use computers these days so be critical in the evaluation of what uses computers – leave only those means of communication which do not use computers anywhere in the line of communication

- Next, have the students discuss what life would be like if they were limited to only the methods of communication which remain on the list. Be prepared to share insight into what it was like to communicate in "the old days."
- Discussion of progression of communication technology and what's next (30 minutes)
  - The goal of this exercise is to get students to reflect on the technology improvements in communications and the rapidly accelerating growth in digital communications in recent decades as well as where that growth is headed in the future.
  - Have the class, individually, in groups, or together as a class, build a list of communication methods throughout history. Do your best to assign rough dates to those innovations. The actual dates of invention are less relevant than the overall result which will likely end with few inventions prior to the late 20th century (for example, speaking, body language, handshakes, newspaper, telegraph, books, etc were from "the dawn of time" until the late 20th century) and significant inventions since the late 20th century (cell phones, computers, internet, facebook, Snapchat, texting, Instagram, email, countless apps and other means of modern communication). Point out to them that the rate of innovation is accelerating and will likely continue to accelerate for the foreseeable future.
  - Have the students gather in groups and discuss these trends and come up with their ideas for where technology will go next in terms of innovative ways to communicate. Have each group share their thoughts informally with the class. Follow up with any thoughts you have regarding where communication technology is headed.
- Email your peers exercise (60 minutes)
  - The purpose of this exercise is to get students to think about the privacy (or lack thereof) of electronic communications.
  - First, hold a class discussion on the importance of professional and acceptable behavior on the internet. There are new examples every day. The chairman of Papa Johns got in trouble for using the N-word and lost his job and the company lost millions in sponsorships. Roseanne Barr had her entire sitcom show cancelled because of a tweet that contained racist language. You can find other examples by searching on Google "people who lost their jobs due to social media". Talk about some of the examples and get the students to recommend rules for themselves regarding what they should and should not post online.
  - Talk about the internet and trust. Trust can be impacted by the systems (they can be hacked or compromised and you lose the security of your data) and the people (they can post false or erroneous information and they can be trying to deceive you or steal from you). Talk about phishing scams and spam email. Phishing scams are communications, typically emails, from people posing as legitimate companies or honest people conducting innocent business who try to get people to reveal personal information such as passwords or credit card/bank account numbers. An entertaining video on spam is called "This is what happens when you reply to spam email" by James Veitch from TED. The video can be found at: https://www.youtube.com/watch?v=_QdPW8JrYzQ and you'll want to stop the video at about the 7 minute mark to avoid some sensitive material.

- Next, conduct an exercise to illustrate how email works and to help students understand the issues regarding internet privacy:
  - Tell every student in the class to think of another student in the class but not reveal to anybody the name of the other student. Tell them to take a piece of paper and write a short note to the student about whom they are thinking. You may want to remind them to say nice things only and nothing offensive in their note. When the students are finished writing their notes, tell them to deliver their notes but to do so by following these rules:
    - Imagine a path from themselves to the destination student where the path includes "stops" at each student's desk sitting between them and the destination.
    - Pass the note to the destination student by handing the note to their nearest neighbor on that path to the destination student.
    - That student will then pass the note to their nearest neighbor in the path to that student and so on.
    - Students are not allowed to provide any instructions to the other students including the name of the destination student, the path the note should take, etc.
  - You'll likely have a bit of chaos because a large number of your students will likely not have included the name of the student on their note, making it impossible for other students to know where the note is supposed to go.
  - After letting chaos run its course for a bit, throw away all the notes and start over. Tell the students to write a note to the same intended students and try again. This time, they will likely put the destination student name at the top of the note and the notes should be deliverable.
  - Repeat this by letting students try again until every note is successfully delivered.
  - Once all the notes are delivered, tell the students who received the notes to reply to the sender by writing a reply. Have them send it back to the original sender the same way they sent the prior notes. This will likely fail because many of the students likely did not include a "from" on the note. If that's the case, let chaos run its course again for a few minutes and then throw away the notes and start over again. Keep doing this until all student notes are received and replied to. You may want to find different ways to assign students peers to whom they should write simply because if a two people or more send to the same person, some students will be left out of the receive/reply portion of the lesson.
  - Point out to students that these are the parts of an email message that are required (to, from, body, and subject). Explain the CC field and you may want to experiment with that too by incorporating it into the exercise.
  - Once you've gone through this exercise, explain that this is how internet communication works. There is rarely, if ever, a direct link between two devices and instead, when you email, post to Snapchat, or text a friend, the message

goes through any number of hops to get to the destination. Ask the students how that impacts security. Point out to them that along the way in their in-class exercise, all the students along the path of a message could open it up and read the contents. The same is true for digital communications (although most messages are encrypted, nothing is every fully secure and hack-proof).

- Digital Footprint discussion (60 minutes)
  - The purpose of this activity is to get the students to think more critically about the data they generate about themselves both in terms of quantity and context.
  - There are several options to consider for this activity:
    - There are a lot of great videos on the internet that talk about big data and how much data we generate and how that helps us and how it may harm us. A great video (though one with unfortunately obnoxious background music) that lasts nearly an hour is: https://www.youtube.com/watch?v=m9D-v6r3NJQ
    - A series of great Ted Talks on big data and the challenges we face in the digital world can be found at: https://www.ted.com/playlists/130/the_dark_side_of_data
    - Get your class to participate in the NSA Day of Cyber where the class can experience a day in the life of a cybersecurity expert. Details on the program can be found at: https://www.nsadayofcyber.com/
    - Get a data privacy/big data guest speaker. Ideas could include local college professors, local industry experts, or police officers or FBI agents.

Module 1: Day 10

Instructional Day: 10 (Topic description, objectives, and ECS outline can be found on pages 46-53 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

Outline of the Lessons:

- Social media exercise and discussion (30 minutes)
- Data problems discussion and presentations (30 minutes)
- Discuss the students' use of data and production of data (20 minutes)

Student Activities:

- Participate in a social media exercise
- Participate in various discussions
- Participate in a group setting, evaluating data problems

- Present findings from a data problem to the class

Teaching/Learning Strategies:

- Social media exercise and discussion (30 minutes)
    - Have the students spend 5-10 minutes writing down a list of all the things they did in the past two days and tell them to include good things, bad things, frustrating things, positive things, etc. Tell them the list is going to be completely confidential – they will be the only person who ever sees the list so they should be completely honest.
    - After they have created that first list, have them create a second list of all the things they have, or would have, posted to social media over the past two days. Give them another 5-10 minutes to come up with that list.
    - Start a discussion about what differences there are in their lists, without requiring them to get into specifics - let them share specifics only if they feel they want to. Most likely, the bulk of the students will have identified some difficult and/or frustrating times they've had in the past two days but will have only posted positive things to social media. Continue the discussion with questions such as:
        - Would you post a picture of yourself that you feel is ugly? Why or why not?
        - Would you post about a time in your life where you weren't proud of what you were doing? Why or why not?
        - Talk to the students about the fake story of our lives that social media creates. Ask the students how, if they see all their "friends" doing amazing things, smiling, looking beautiful, and having fun, it will make them feel about their own lives? Talk to them about whether or not they feel that's a problem and what they can do about it?
- Data problems discussion and presentations (30 minutes)
    - The goal of this activity is to introduce the students to some interesting and complex problems involving data and get them thinking more strategically about the data they see, consume, and produce. Divide the class into groups and assign each group a problem (examples below) by using some of the resources listed or by finding your own. Give the groups 10-15 minutes to digest the information and then have them present the problem to the class. Things for them to include in the presentation are:
        - Introduce the scenario
        - What type of data is used?
        - How is the data acquired?
        - What are the problems?
        - Can they think of other problems or solutions?
    - Some sample problems for the students to investigate:
        - Strava heat map and the unintentional disclosure of secret military bases:
            - https://www.wired.com/story/strava-heat-map-military-bases-fitness-trackers-privacy
            - https://www.youtube.com/watch?v=0IB8p-YpXwA
            - Sample questions to drive discussion:
                - Was Strava in the wrong to publish the heat map?

      - What considerations need to be evaluated before publishing data?
      - How could this have been prevented?
      - Is there any way that you are creating data which may unintentionally reveal your location or habits to those who may wish to harm you?
  - Alexa murder case and data privacy concerns:
    - https://www.wired.com/2017/02/murder-case-tests-alexas-devotion-privacy/
    - https://www.youtube.com/watch?v=jCtU5jZNMw8
    - Sample questions to drive discussion:
      - Should Amazon have to provide investigators with the Alexa data or not? Why or why not?
      - If Amazon was to provide the data to investigators, what precedent does that set for data privacy in the future?
      - What other ways are data collected that may be compromising to the user?
  - Germany wins 2014 World Cup by leveraging big data
    - https://www.chegg.com/homework-help/questions-and-answers/germany-wins-world-cup-big-data-side-stunning-display-talent-resilience-teamwork-germany-w-q28734544
    - https://www.youtube.com/watch?v=lvwq51p1Sv4&t=21s
    - Sample questions to drive discussion:
      - How can big data be used in sports?
      - Is it cheating to use big data to the level used by Germany in 2014?
      - What is the future of sports as big data usage becomes more and more mainstream?
- Discuss the students' use of data and production of data (20 minutes)
  - Hold a discussion about data and how the students are producing their own data. When do they produce data? Help them see that when they turn on their phones, install an app, drive a car, order something online, checkout at a grocery store, etc, they are creating data. Discuss how they feel about that data they produce; does any of it alarm them or excite them for any reason? Discuss the tradeoff of privacy – the more they share, the better technology is (e.g., many traffic map programs such as Google Maps depend on user data to identify slowdowns and traffic concerns) but the more they share, the less privacy they have (e.g., by sharing that traffic data with Google, the people at Google, and possibly others, know where they are and when/where they drive)
    - This may be a valuable article to peruse prior to class: http://time.com/4673602/terms-service-privacy-security/
  - Prepare the students to complete the homework from ECS in which they record all the data they create. A follow-up homework assignment could be to have the students look

at the data they found they generate and to create a story about themselves that somebody might create if they did not know the student but saw only the data the student created. Would somebody be able to deduce from the data where the student went to school? What route they took to go to and from school? Where they worked? Where they lived? What they like to do? What they like to eat? And… how does the student feel about the story that may be resulting from the data they are generating? Do they feel they need to make any changes to their data production? Why or why not?

Module 1: Days 11-14

Instructional Day: 11-14 (Topic description, objectives, and ECS outline can be found on pages 54-66 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

Outline of the Lessons:

- Introduction to computer graphics discussion (30 minutes)
- Graphic editor tutorial (30 minutes)
- Discussion of image formats (20 minutes)
- Hidden messages activity (60 minutes)
- CGI discussion and activity (40 minutes)

Teaching/Learning Strategies:

- Introduction to computer graphics discussion (30 minutes)
    - The purpose of this section is to introduce students to very basic concepts regarding how computers use video to show pictures and create digital graphics.
    - Watch the following video with the class: https://www.youtube.com/watch?v=15aqFQQVBWU
    - Check with the students and make sure they understand the concept of pixels. If they do not, continue to help them understand. Talk to them about the idea that because the human eye cannot see the detail level of a pixel, those tiny little dots are merged in your brain to look like one continuous image. Help them understand the concept of RGB (red, green, and blue) and that you can take those three colors in different quantities and combine them to create thousands of colors.
    - You can open Paint or a similar program on a computer and play with the RGB color picker to show the students different combinations of colors you can create by using different RGB values.

- Help them understand the concept of screen resolution and pixel density. Essentially, resolution is how many pixels there are horizontally and vertically on the screen (see the previously listed video) and density represents how tightly packed those pixels are. Ask the students which is better, higher resolution or higher density. You can even divide the class up and have a debate, have each side argue why one would be preferred. The answer ultimately is, it depends. Higher resolution means you have more pixels and therefore more detail. However, density matters: the tighter the density, the better images on the screen will appear up close. You could hold a discussion about the difference between large jumbotron screens at sports events vs. cell phone screens. Which need higher resolution and why? Which need higher density and why?

- Graphic editor tutorial (30 minutes)
  - The purpose of this section is to familiarize the students with a basic photo editor. There are a lot of free software tools available for photo editing. Paint comes installed on most Windows computers. Other free tools we'd recommend include GIMP and Paint.NET. We recommend either watch some videos yourself and then teach the students or watch some videos in class. Here are some ideas:
    - GIMP
      - https://www.youtube.com/watch?v=Q8C0LJPpr64
      - https://www.youtube.com/watch?v=rUKbD1hhEEM
    - Paint.NET
      - https://www.youtube.com/watch?v=l_F0XHACU1Q
      - https://www.youtube.com/watch?v=QBY5_d8uwro
- Discussion of image formats (20 minutes)
  - The purpose of this section is to help students understand a few of the pros and cons of different image formats. You'll notice when saving images in an editor, you're presented with a variety of options regarding file format. Is there a difference? Yes. Discuss some of those differences as noted in the following video:
  - https://www.youtube.com/watch?v=MnIUBLxNzNM
- Hidden messages activity (60 minutes)
  - The purpose of this section is to give the students some experience editing images in an in-class activity.
  - If you have a large enough class, you may want to divide the class into teams or groups. Each group is to edit an image. You can assign the images (recommended) or let them find an image on the internet. We'd recommend selecting images yourself from something like a Google search for "scenic images" (https://bit.ly/2nhFGMx). Regardless, make sure the image you select is high enough quality that it looks crisp and photo-quality to the eye. Each group will then open their image in a photo editor. Have the students zoom in on the image enough that they can see the individual pixels. Help reinforce the idea that the human eye, when far enough away from the pixels, blends those pixels to make a continuous, crisp image.
  - Each team is to take the image and find a way to put a hidden message in the image. You may want to provide the message to the students so they all have similar length of messages to hide within the image, but that's up to you. You should walk them through

doing this one time and help them see the difference when using different colors for their hidden messages, etc and how those do or do not blend into the background. Here is an example:

- Original image:

- Image with hidden message (The message is on a tree trunk just to the right of the road):

- Zoomed in on the message:

  - Once each team has edited an image and experimented with embedding a hidden message, provide them with a new image and message and a task to conceal a message that will not be found by their peers. Give them 20 minutes or so to get together and hide their message. Then have the teams give their images to another team and give that team some time to find the message. You could have two teams swap images and have them race to see who can find the other team's message the fastest. You could even repeat this task and have some kind of tournament to find out who can best conceal the messages within the images.
  - You may want to consider having the teams turn in their image with instructions on how to find their hidden message as an assessment.
- CGI discussion and activity (40 minutes)
  - Now that the students have had some experience editing images, hold a discussion about CGI. Talk to the students about how tedious and complicated it is to edit images by hand the way they just did with the hidden messages. Ask them to think of a profession that uses a lot of image editing. Get them thinking about films/television and CGI. CGI stands for "Computer Generated Imagery" and is the primary means of special effects in Hollywood today. A good example of how times have changed in Hollywood that the students can likely relate to is the original Star Wars compared to the current ones. In the original films, many of the aliens and odd creatures were actual humans or animatronics (robotic puppets). A great, albeit long, video on how the original Star Wars creatures were created can be found at: https://www.youtube.com/watch?v=AWC7nDX055c. At the 36:28 mark in the video, they talk about how they created the original Jabba the Hutt, which is an interesting clip to show if you don't want to have the class watch the entire video. You can show clips of the new Star Wars movies to show the class examples of CGI Star Wars effects. Hold a discussion of the pros and cons of the animatronics or other low-tech methods vs. the new CGI effects. Talk to the class about how the CGI effects are done. Essentially, they are similar to editing the images like the class did with hidden messages, but thankfully editors have better tools at their disposal. Essentially, they are going frame by frame, however, and editing images in a similar manner.

- If you have enough time, you can get into a discussion about CGI-generated images of people who aren't even acting in a film. A couple of specific examples you could use revolve around Star Wars once again (what doesn't?).
  - Grand Moff Tarkin (alive during the filming of A New Hope but deceased and CGI in the filming of Rogue One).
    - https://www.youtube.com/watch?v=KsuvXHGCVXE
  - Princess Leia (young Leia was completely CGI at the end of Rogue One)
    - https://www.youtube.com/watch?v=6Yj31YCa3Xw
    - Leia is also expected to play a major role in Star Wars Episode IX which is expected to be released in December, 2019. However, she passed away before the majority of the production took place in the film so she will likely be a largely CGI-based character in the movie.
  - Discussion: What are the pros and cons of being able to so accurately depict people in CGI? Can the students think of any negatives?
  - You could also get into discussions about CGI in video games. There are a vast number of CGI examples and "how did they do it" videos available on Youtube that you may find beneficial.

Module 1: Days 15-16

Instructional Day: 15-16 (Topic description, objectives, and ECS outline can be found on pages 67-70 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

Outline of the Lessons:

- Human PacMan Game (60 minutes)
- Programming PacMan in Rules (60 minutes)

Teaching/Learning Strategies:

- Human Pac-Man Game (60 minutes)
  - The goal of this section is to help the students understand on a rudimentary level how computer programs work including timing and the issuing of instructions. Plan ahead and have a life-size Pac-Man map arranged for the students to use. You can rearrange desks in the classroom to make the maze or put tape on the floor or anything else you feel would work. If you're not familiar with Pac-Man, you can play the game here: http://www.freepacman.org/. You may also want your students to play it a few times to

get the feel for it before participating in this activity if they are unfamiliar with the game.

  - In the life-size Pac-Man map, you're going to have the students play Pac-Man. You'll have to find something to represent the dots that Pac-Man eats as well as the flashing power pellets that let Pac-Man chase and eat the ghosts. These could be simply pieces of paper that you place on the ground around the maze or really anything else that you think might work.
  - Assign starting points in the maze for Pac-Man as well as a ghost center box which serves as a starting point for the ghosts as well as a return area when they are eaten. Select a student to play as Pac-Man and four others to play as ghosts and have them go to their starting locations.
  - Simulate instructions that are processed by a program by letting the players each move one time in a computing "cycle" and wait for all players to have completed their moves before moving on to the next cycle. In a given cycle, let Pac-Man move two steps in whatever direction he/she chooses and let each ghost move 1.5 steps in any direction they choose.
  - Play through a few games so students get the feel for the game. You can swap out participants each time through the game to give others a chance to play.
- Programming Pac-Man in Rules (60 minutes)
  - In this second half of the Pac-Man lesson, students will program the ghosts and the Pac-Man by coming up with predetermined instructions for each. Divide the class into teams. Each team will create a set of instructions for the ghosts to follow and another set of instructions for the Pac-Man to follow. Each ghost can be programmed separately (yielding 5 sets of instructions, 4 for the ghosts and 1 for Pac-Man) or all the ghosts can follow the same instructions (yielding 2 sets of instructions, 1 for the ghosts and 1 for Pac-Man).
  - Sample instructions could be:
    - Ghost:
      - Move forward if possible.
      - When not possible, turn right 90 degrees, then move to step 1.
    - Pac-Man:
      - Move to the space furthest away from all ghosts
      - In case of a tie between choices, take a random choice from among the tied choices
  - Another sample instruction set:
    - Ghost:
      - Move to the space that takes you closer to the Pac-Man
      - In case of a tie between choices, take a random choice from among the tied choices
    - Pac-Man:
      - Move toward the nearest dot
      - In case of a tie between choices, take a random choice from among the tied choices

- Once all the teams have created their rule sets, rotate through the teams and let each team's Pac-Man play against the other teams' ghosts. Track which teams perform the best to create a competition among the students.
- Talk to the students about how this relates to programming. The essential point to get across is that the computers do not think well on their own. Instead, computers are exceptional at doing exactly what you tell them to do. Even though it might make sense to us to eat the dot right in front of Pac-Man, if the program instructions are to go the other way, the program will do exactly that.
- You may want to consider having students turn in their rules as an assessment.

Module 1: Days 17-19

Instructional Day: 17-19 (Topic description, objectives, and ECS outline can be found on pages 71-73 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

Outline of the Lessons:

- What is intelligence? (60 minutes)
- Chat with bots (60 minutes)
- Test bots & rework with themes (60 minutes)

Teaching/Learning Strategies:

- What is intelligence? (60 minutes)
  - Hold a class discussion on intelligence. What is it? How do we define it? Start by talking about human intelligence. How can you define that? There are a bunch of different ways we try to quantify it (grades, test scores, IQ tests, etc). What are the pros and cons of these methods? Do they do a good job of defining intelligence? Why or why not?
  - Watch the following video to get the students thinking more broadly about intelligence: https://www.youtube.com/watch?v=9xTz3QjcloI. The students will likely be overwhelmed by the content in the video as there's a lot to process and it's complicated. That's kind of the idea, though. Intelligence is difficult to explain and this video helps them see that even the worlds foremost experts struggle to define intelligence.
  - After the video, restart the intelligence question. Remind them of Thurstone's clusters of mental abilities:
    - Spatial ability
    - Verbal comprehension

    - Word fluency
    - Perceptual speed
    - Numerical ability
    - Inductive reasoning
    - Memory
- Ask the students how to define intelligence if there are so many different components that contribute to the concept of intelligence. Does intelligence mean you're good at some of those abilities or all? Does intelligence mean that you're good at something or that you have the potential to be good at it (e.g., you can learn).
- Remind the students that many other experts had other ways to define the idea of intelligence. What about emotional intelligence? How does it differ from intelligence in general? The essence that the students should understand: intelligence is difficult, maybe impossible for us to define.
- Now move the discussion toward computers. Talk about the ideas that Hollywood puts out there in terms of artificial intelligence. Things like Ironman's Jarvis, the Terminator in the Terminator movies, and HAL 9000 in 2001: A Space Odyssey. Hollywood painted this picture of artificial intelligence and the dangers of the computers becoming so intelligent that they pose a risk to humanity (Terminator, The Matrix, etc). Have the students offer some opinions on how we should define artificial intelligence.
- It will likely be difficult for the students to define artificial intelligence. Ultimately, that's because it's hard to define actual intelligence (as they just found out). If we can't even define intelligence, how can we determine whether something that is supposed to artificially mimic intelligence does it effectively? Help students understand that this is the fundamental issues that has always surrounded the field of artificial intelligence – and the debate is still ongoing today!
- Watch the following video with the students to introduce the concept of the Turing Test to determine artificial intelligence: https://www.youtube.com/watch?v=sXx-PpEBR7k
- Discuss with the students whether or not the Turing Test is a good test for intelligence. There really are no wrong answers because we still don't really even know if that's a good measure of intelligence again… because we can't really define intelligence ourselves.
- Talk about interesting advancements in this area. Students may have some ideas on their own. Here are a couple of others:
    - This video (https://www.youtube.com/watch?v=WbCguICyfTA) talks about a teacher at Georgia Tech who developed an AI bot (a chat program) that worked as the TA in his class. The AI bot was called Jill Watson and the program actually won the award for the best TA that semester and a male student actually asked Jill out on a date. Talk to the students about the fact that this entire experience was a giant Turing Test. What are the pros and cons of using programs that can so effectively pass the Turing Test?
    - Google has a new AI program that can talk more realistically than perhaps any program before it. See more about it here: https://www.youtube.com/watch?v=lXUQ-DdSDoE What are the pros and cons

of having programs such as this which are so hard to determine whether they are people or computers (which again, is a giant Turing Test!)?

- Chat with bots (60 minutes)
- Test bots & rework with themes (60 minutes)

Module 2: Days 1-2

Instructional Day: 1-2 (Topic description, objectives, and ECS outline can be found on pages 76-78 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

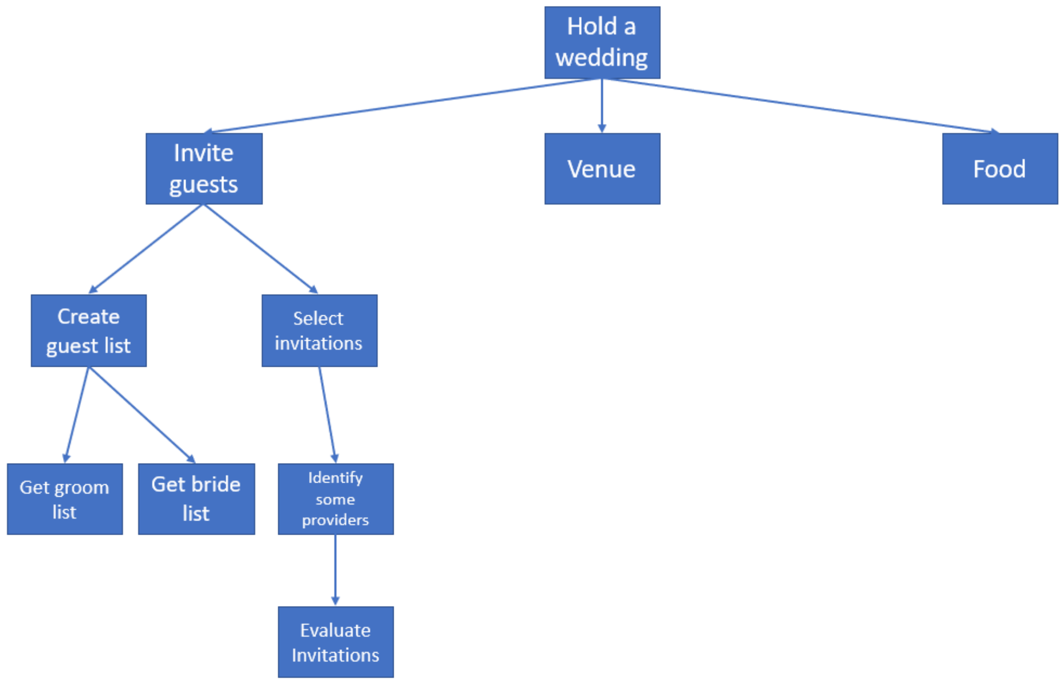
This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

Outline of the Lessons:

- Problem Decomposition (60 minutes)
- Problem Decomposition Activity (60 minutes)

Teaching/Learning Strategies:

- Problem Decomposition (60 minutes)
  - The purpose of this section is to help the students understand the rationale behind problem decomposition and how it is done.
  - It is helpful to start this discussion by holding a discussion with the students about tackling a large problem. A good place to start is to introduce a large problem to the students. Some ideas could include:
    - Imagine you're a bride or groom and you're planning your wedding
    - Imagine you're the commissioner of the NFL and you're planning the next year's season
    - Imagine you're the architect/planner of the school building and you need to build the school
  - Hold a discussion with the students regarding how they would accomplish one of these or another large project. The discussion will likely automatically turn into the breakdown of the task into smaller tasks, such as breaking "hold a wedding" down into "invite guests", "contract a photographer", and "buy a wedding cake." This, ultimately, is the essence of problem decomposition. However, there is a programmatic approach to it that can be useful if problems are broken down strategically into smaller and smaller problems in a tree structure or other similar representation:



When breaking down problems in this manner, the arrows represent dependencies. To learn more about problem decomposition, you can check out some of the following resources:

- https://www.youtube.com/watch?v=yQVTijX437c
- https://www.youtube.com/watch?v=jJB8JHv5VGI&t=17s

To see how you would use this in a computer programming scenario, watch: https://www.youtube.com/watch?v=ZgPltuFFrno

- After the discussion, the students should have a solid understanding of why it's important to break down tasks into smaller, more manageable tasks. Now, let's try a simple task. Before the lesson, watch the following Ted Talk: https://www.ted.com/talks/tom_wujec_got_a_wicked_problem_first_tell_me_how_you_make_toast
- You're now going to have the students diagram how to make toast. First do this individually, on a piece of paper just like the speaker did. Next, have them form groups and use sticky notes to first come up with an individual representation and then merge that representation with the rest of their group to have one, detailed but concise, diagram for the group using sticky notes (as shown in the Ted Talk video). If you feel it would benefit the class after the activity to watch the Ted Talk video, show it to them.
- Wrap up this section with a brief discussion regarding the benefits of breaking down tasks and representing them in smaller, more manageable tasks. At the end of the Ted Talk video, the speaker talks about some other key reasons why breaking down tasks is important (things like, aligning people to a common vision and understanding of the project and goals). These concepts should come out in the discussion as benefits as well.

  Other possible resources:

  - http://www.bbc.co.uk/guides/z8ngr82

        - https://www.youtube.com/watch?v=jJB8JHv5VGI
- Problem Decomposition Activity (60 minutes)
    - In this section, you'll have the students practice work breakdown problem decomposition. Students will be breaking down a large project into manageable steps. This can be an individual or group exercise, but to facilitate larger projects and to get the benefits of idea-sharing and brainstorming, groups are recommended. Have each group of students select a large project to break down. Encourage students to find something relevant to them. Problem examples might include:
        - How do we get better food options at the school's football games?
        - How do we make registration for school classes easier?
        - How can we increase recognition for student academic accomplishment?
    - Have the students work on their problems and break them down into smaller problems. For problems like these, help them understand the concept of stakeholders and people who's buy in are required to move forward. Students should design those into their breakdown by creating tasks to gather information, solicit opinions, or do research to present opportunities to stakeholders. These problems may lay a foundation for the student projects they will complete at the end of Module 2, so you may want to help students select projects that can help them in that regard.
    - After the students complete their breakdowns, you can have some or all of the teams present their breakdown, if you have time and it makes sense for the flow of the class.

Module 2: Day 3

Instructional Day: 3 (Topic description, objectives, and ECS outline can be found on pages 79-81 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

In our design meetings with Utah ECS I teachers, the consensus from the teaching community was that this lesson in the ECS I curriculum works great and did not need additional resources. What's working for you in your classroom? What kind of help would you like to see? Post a comment to discuss…

Module 2: Days 4-6

Instructional Day: 4-6 (Topic description, objectives, and ECS outline can be found on pages 82-86 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

Outline of the Lessons:

- Introduce the problem (30 minutes)
- Client/server discussion (30 minutes)
- Groups/individuals work on their pieces (60 minutes)
- Groups/individuals consolidate everything and rework (30 minutes)
- Present and share (30)

Teaching/Learning Strategies:

- Introduce the problem (30 minutes)
    - The goal of this lesson is to get the students to apply some problem-solving thinking via a "how did they do it?" type of problem. We recommend doing so using something technical with which they are familiar. Fortnite (https://www.epicgames.com/fortnite/en-US/buy-now/battle-royale) is a very popular video game right now and that's where we'd recommend you start. If you or your students are unfamiliar with Fortnite, you can view some gameplay here: https://www.youtube.com/watch?v=oJYtauvqI3Q. Alternatively, if you don't want to use a video game in this problem, pick some other high tech system, ideally one with many moving parts (so it's more fun and challenging to dissect). You could take a scene from a popular movie, for example: https://www.youtube.com/watch?v=I7mDue5HOE8. But we do believe the video game concept will work the best for this exercise. For clarity, the rest of this lesson will describe the project in terms of Fortnite.
    - Introduce the problem to the students: they are going to identify all the components involved in a Fortnite game and come up with rules for how they might work. When building software, you typically start with some sort of requirements document which describes how the system you're going to build is intended to work. That would be similar to what the students will create here. If you're not too familiar with how to do this, don't worry. There aren't many wrong answers because, there are a lot of different ways you could create something like Fortnite. Dive in and experiment with the students and help them open up their creativity as they look at this problem.
    - Inform the students that they will break down the game into as many distinct pieces (components) they can and build rules for each (because we're reverse engineering Fortnite, we'll call the process of building rules for a component "breaking down" that component. Here are some guidelines they should follow as they identify the different components:

- All components will be broken down as follows:
  - Name (give the component a name)
  - Description (short description of the component)
  - Created by (how and when is this component created)
  - Killed by (how and when is this component deleted)
  - Video (what does the component look like, if visible at all)
  - Audio (what does the component sound like, if audible at all)
  - Action (what action(s) does the component take or what events does it create)
- Help the students understand that everything in Fortnite (and most computer systems) has a life span. Things are created, they act, and they go away. This is similar in all programs (think of when you're copying files on a computer. One similar component that has a life span is the pop up box that shows the progress of copying those files. It is created when a copy is started. It lasts until the copy is done. While it's "alive" or "active", it shows the user how long the copy has lasted and how long it has left. Components in Fortnite are no different. Some are created when the game starts (players, trees, houses, etc). Some "die" at different times (players when they are killed, furniture when the player breaks them, bullets when they hit an object, etc). And all act differently (trees stay in one place but may sway in the wind, players move as directed by the user, etc).
- Everything that moves (or acts) independently should be identified as a component to break down. This includes the players, the weapons, ammo, parts of buildings and the environment, the sky, the storm, the music, etc).
- Things that are the same, or that entirely act the same, should only be identified and broken down once (for example, even though there are different models for trees (some trees look different from others), they all behave the same so you only need one component for "trees". The same is true for "walls" and "furniture".
- Even though players act generally the same, they should be broken out into two components, "my player", which represents the player being controlled by the person playing the game on that computer or console and "other players", which represents all other players in the game. More on this in a minute.
- Some things change state and behavior throughout the game. Ammo, for example, is available in "loot" on the ground that you can pick up. Once you pick it up, it doesn't show up on the screen again until you shoot it and, when you shoot it, it behaves very differently than before you picked it up (before you pick it up, it's spins on the ground and glows but it does not move throughout the game world, after you shoot it, it flies through the air looking to create some damage). Each state should be identified as a component and broken down separately. Ammo, for example, really has three states and therefore should be broken down as three components: loot (spinning on the ground, waiting to be picked up), on-person (after pick up but before being shot, not visible other

than as a total count of bullets available), in air (flying through the air, dies when it hits something).
        - Everything that you can see, hear, or that acts in some way in the world should be included as a component. This includes things you may not think about such as gravity, the different parts of the HUD (heads up display) like the player's health bar, enemy health bars, the inventory slots, the minimap, the large map, wind, the school bus at the beginning, etc.
        - Games are typically created in a "scene" format where there is a world that does not move and there is an invisible camera floating in the world which moves based on the user's command (turns, looks up and down, and moves forward and backward). Help the students understand this part of the game - imagine a giant model you've created and placed on a table or the floor and a camera that you've moving through the model – that's how games are made. So, when it comes to the "world moving" when you turn around as a player, help the students realize that it's not the world moving, but the camera within the world is rotating. (and yes, that means "camera" should be a component the students break down).
- Client/server discussion (30 minutes)
    - Start a discussion with the students about players regarding client/server architecture at a very high level. Don't worry if you aren't sure what that is, you'll understand enough about it to have a discussion with the class after reading this.
    - Help the students imaging a game like PacMan. It's a single-player game played on a single computer. How does the game know where all the players are and what they're doing? Ask the students… they'll come up with interesting ideas.
        - Answer: As a single player game, the ghosts in PacMan are controlled by the program and it tells the ghosts where they should be and what they should be doing.
    - Now imagine something like Madden (a popular football video game: https://www.youtube.com/watch?v=8hBj7_yXYuY) when its played in multiplayer mode but only on one computer (ie, sitting on your couch with your friend, both of you have a controller, and you're playing against each other. Ask the students again how the system knows where all the players are and what they are supposed to do.
        - Answer: Here, you'll have 22 players, 11 on each team (let's only talk about those that are on the field, even though you can see others on the sidelines). 2 of those 22 players are controlled by the 2 players and the computer knows where they should be because it knows where they were a few microseconds ago and where they've gone since then. However, it doesn't know what they'll do next – that is dictated by the players. So the players send input to the computer via the game controllers and the game processes that input and makes the player act accordingly. The other 20 players are controlled by the system and are essentially the same as the ghosts in PacMan – the system knows where they are and what they'll do next.

- Now, ask the students how the computer will track players in Fortnite. The difference here is that you're not all at the same computer. There are no computer-controlled players, they are all controlled by people, but each player has their own computer. How would the students make that work?
    - Answer: one way the students may have identified as a possibility is have each computer send updates to all the other computers telling them where their player is. This is called a peer-to-peer architecture and yes, it's a real thing. However, it's very inefficient for problems like this. Imagine 100 computers with 100 players. Each computer will send an update every few microseconds to every other computer and process them from every other computer as well to track all the other players itself. That's a lot of messages floating around! Instead, we use something called a client/server architecture. In this case, the server tracks the status of everything that needs to be shared throughout the game. Each of the user computers are called clients. The receive messages just as frequently as the example in the peer-to-peer architecture (every few microseconds) but they only receive one message – from the server. They also publish updates at the same rate, but only one – to the server. So, effectively, when a player moves forward on his controller, they are sending input to the computer, the computer processes that and moves the player forward, the computer then sends a message to the server telling it that it moved the player forward and now the player is at a new position, the server then sends that updated position to all the other clients, the other clients receive that update and update the opposing player's position accordingly.
- Why is this important? Aside from being a critical piece of computer architecture, this will be key in the students' dissection of the game. As they come up with the actions of each of their components, they should identify when to send updates to the other computers, via the server, and what those updates should contain.

- Groups/individuals work on their pieces (60 minutes)
    - Divide the class into groups. Have each group work to identify the components in the game together. Then, when a group feels they have cited all the components, have them make assignments within the groups to subgroups (1-3 students) to breakdown the components as outlined above.
- Groups/individuals consolidate everything and rework (30 minutes)
    - After the smaller subgroups have completed the breakdowns of their assigned components, have the larger groups get back together and review everything their subgroups created in their breakdowns. Have the groups make any final adjustments they feel necessary.
- Present and share (30)
    - Select some or all groups to present their components and breakdowns to the class.
    - The final breakdown of all components would be a good assessment to have the students complete and submit.

Module 2: Days 7-9

Instructional Day: 7-9 (Topic description, objectives, and ECS outline can be found on pages 87-88 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

Outline of the Lessons:

- Problem solving through code (150 minutes)
- Present and share (30)

Teaching/Learning Strategies:

- Problem solving through code (150 minutes)
  - The goal of this lesson is to let the students solve a large problem by using technology and code. The problem: build a website to store all the information from the project the students completed in the previous lesson (or some of the information, if the end result was too large and was mostly hand-written).
  - Have the students first draw out what they want the website to look like on a piece of paper. Websites should be simple and include things like an image header, a title, various level headings, and text. If the students are capable of more complex websites or simply want to try and extend themselves, feel free to let them go as complex as they want.
  - Point the students to the W3 Schools HTML pages where they can learn how to create a website in HTML (https://www.w3schools.com/html/default.asp). Let the students work in their teams or individually, depending on the size of the groups – you just want to make sure the students are working in small enough groups that they are all learning and producing rather than having some sit by the side and not participate. Give the students enough time (2.5 hours at least is recommended) to learn and build.
- Present and share (30)
  - Have the students present their websites to the class afterward. You may want to have them talk about how they built their websites, particularly if there is a student or group that adds something more complex into their site.

Module 2: Days 10-12

Instructional Day: 10-12 (Topic description, objectives, and ECS outline can be found on pages 89-90 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

Outline of the Lessons:

- Binary number system (60 minutes)
- Binary messages project (60)
- Binary images project (60)

Teaching/Learning Strategies:

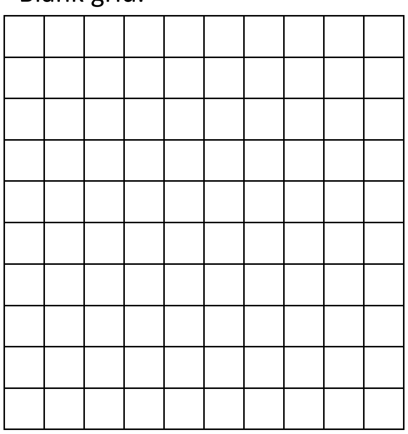
- Binary number system (60 minutes)
    - The goal of this lesson is to help the students understand how the binary number system works. For starters, this video (https://www.youtube.com/watch?v=LpuPe81bc2w) does a good job describing how binary works.
    - Next, this video does a great job of taking it to the next step and walking through an example: https://www.youtube.com/watch?v=H4BstqvgBow
    - Continue throwing examples to the students until you feel they have a good understanding of how it works. If you have time, you could look at making a game out of it and having the students work in teams, family feud-style where you give a binary number or decimal number to one student in each team and see who can convert it to binary/decimal the fastest.
- Binary messages project (60)
    - Help students understand that everything in a computer is stored in binary because, at its core, a computer is a series of electronic circuits that are either on (1) or off (0). That means that when you write and save a document in Microsoft Word, the computer translates the text into binary and stores it. When you open the Word document again, it translates it from binary back to text. Start a brief discussion with the students by asking them to think about if they were designing a computer, how would you represent text from binary. See what answers the students come up with.
    - Introduce the concept of the ASCII chart to the students. Work on ASCII began in 1960 and it was created to solve this problem: how to represent characters (text) in binary. Essentially, the ASCII chart is just a lookup table where you take a number (represented in decimal, binary, hex, or any number system, really) and map it to a text-based character. A great ASCII chart can be found at: http://web.alfredstate.edu/faculty/weimandn/miscellaneous/ascii/ascii_index.html
    - Using the chart, you can see that the binary code for the uppercase letter 'H' is 01001000. You may want to walk the students through how to look that up. Then you may want to have them convert that number to decimal just for more practice. You can

also see that the binary code for the lowercase letter 'i' is 01101001 and the binary representation of the exclamation point is 00100001. Help the students see that when you type "Hi!" in a text file and save it, the computer is actually going to store those binary representations in memory so "Hi!" will be stored as 010010000110100100100001. It's common notation with binary numbers to break them out into 8 or 4 digit pairs when writing them just to keep it less confusing so you can write that sequence instead as: 0100 1000 0110 1001 0010 0001 but it is important to note that the computer just sees it as one big string of binary digits.

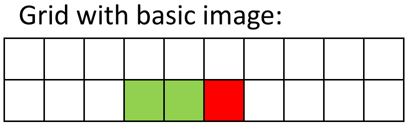
  - For the remainder of this project, have the students pair off and have each student write a note to the other in binary. Then have them pass the notes to the other and have them decrypt the notes. You may want to have them convert the binary to decimal in the process just to continue practicing binary conversion.

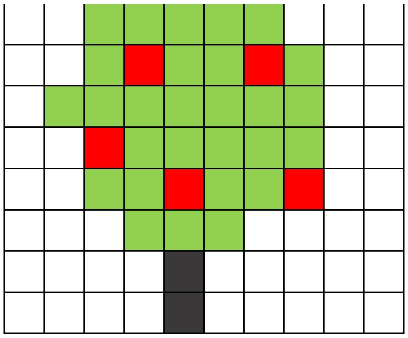
- Binary images project (60)
  - Hold another brief discussion with the students about color. Every pixel on the monitor can be represented by a color and in computers, those colors are typically created using an RGB notation. RGB stands for Red, Green, Blue and the typical format for those colors is two hexadecimal characters for each component, red, green, and blue. If you have time to introduce hexadecimal to the students, go ahead and teach RGB using hexadecimal. Otherwise, help students understand that the RGB components also map to decimal and have a value range of 0-255 for each red, green, and blue component. A good color picker chart can be found at: http://doc.instantreality.org/tools/color_calculator/
  - Have the students draw a very simple image using a small grid such as a 10x10 grid:

Blank grid:



Grid with basic image:





- Next, have the students assign a color in RGB format (decimal is fine) for each of the cells in their image. Then, have them convert those decimal RGB notations to binary. Each component in RGB (the red, the green, and the blue) will convert to 8 binary digits. So, each cell will be represented by 24 binary digits, which is three 8-digit binary numbers consecutively. Finally, have the students write out the representation of the image in one giant binary string. As they see that string, it will be incredibly long, help them realize that when you have an image as simple as the one shown, that's what the computer stores in memory. Imagine how big the binary representation of a photo-quality image would be!
- Next, have the students pass their binary representation of their image to a partner and see if that partner can translate it back to an image. See if they get the same image as their partner had originally.

Module 2: Days 13-14

Instructional Day: 13-14 (Topic description, objectives, and ECS outline can be found on pages 91-93 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

In our design meetings with Utah ECS I teachers, the consensus from the teaching community was that this lesson in the ECS I curriculum works great and did not need additional resources. What's working for you in your classroom? What kind of help would you like to see? Post a comment to discuss…

Module 2: Days 15-16

Instructional Day: 15-16 (Topic description, objectives, and ECS outline can be found on pages 94-95 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

In our design meetings with Utah ECS I teachers, the consensus from the teaching community was that this lesson in the ECS I curriculum works great and did not need additional resources. What's working for you in your classroom? What kind of help would you like to see? Post a comment to discuss…

Module 2: Day 17

Instructional Day: 17 (Topic description, objectives, and ECS outline can be found on pages 96 of the ECS 8 curriculum)

Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

In our design meetings with Utah ECS I teachers, the consensus from the teaching community was that this lesson in the ECS I curriculum works great and did not need additional resources. What's working for you in your classroom? What kind of help would you like to see? Post a comment to discuss…

Module 2: Days 18-21

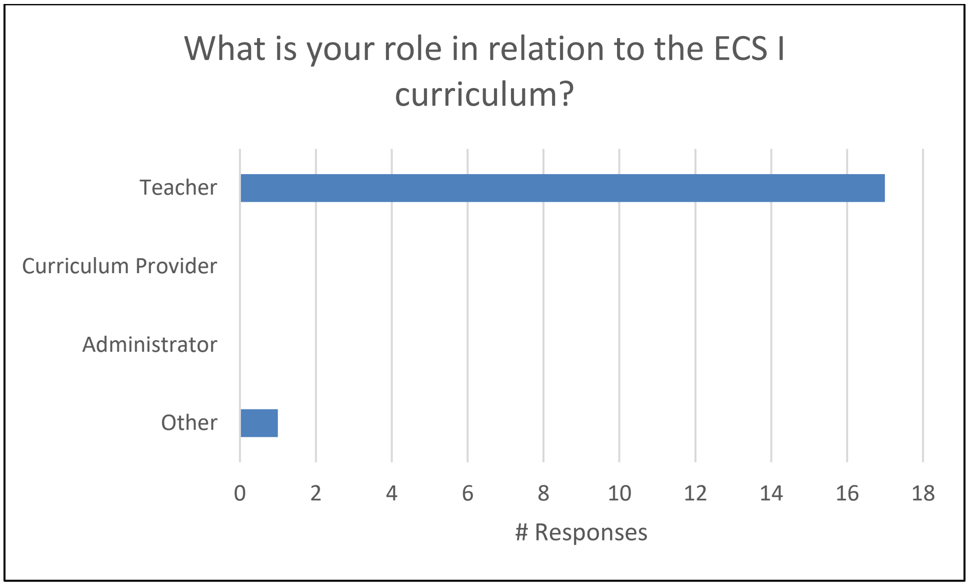
Instructional Day: 18-21 (Topic description, objectives, and ECS outline can be found on pages 97 of the ECS 8 curriculum)

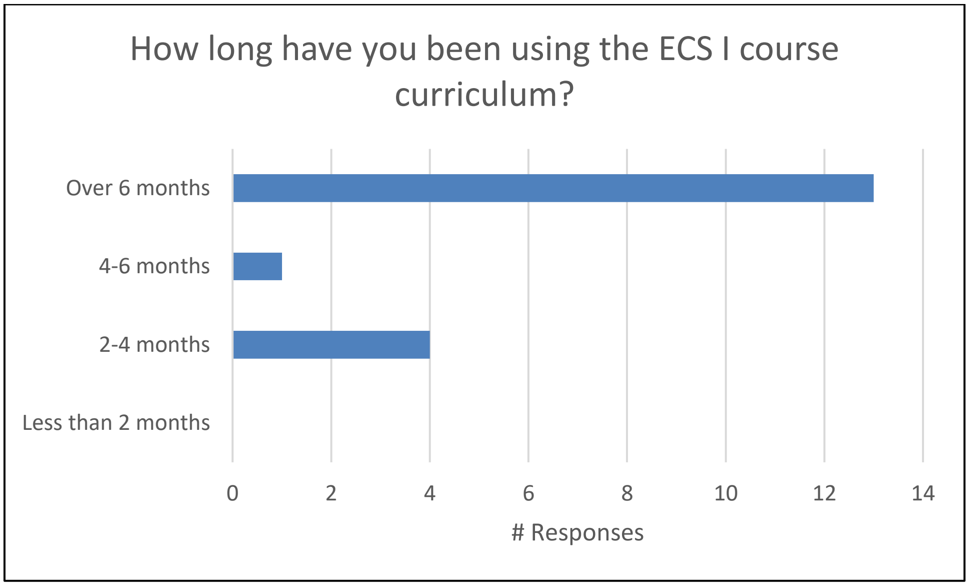
Note: This lesson plan is meant to serve as a supplementary resource to the existing Utah ECS I course curriculum which is based on the Exploring Computer Science curriculum and can be found at: http://www.exploringcs.org/.

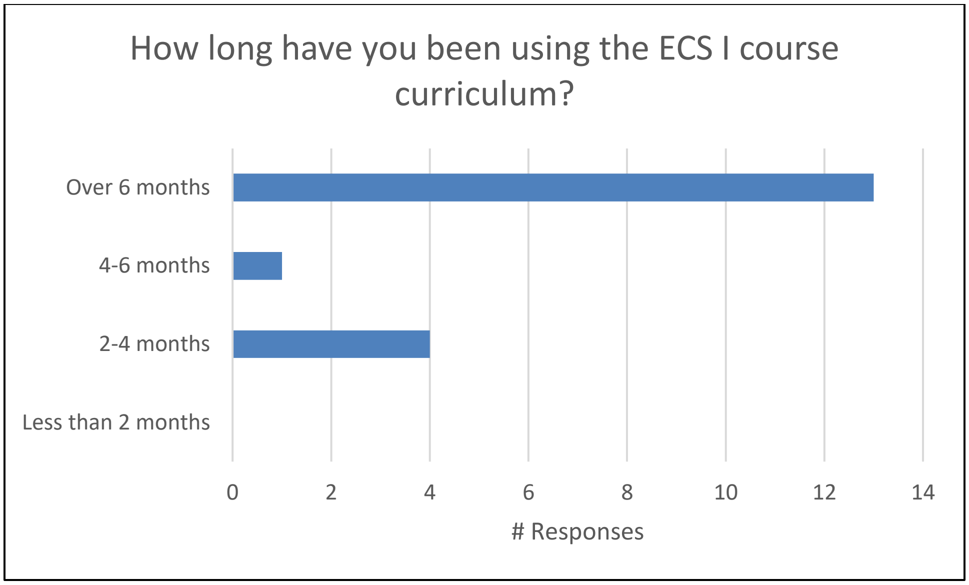
This lesson plan is not meant to replace the existing Utah State ECS I course lesson plan but instead to provide Utah computer science teachers with alternative ideas for lesson plans which can be used to teach the lesson objectives. Teachers using this resource should be aware of their overall course and lesson objectives as outlined by the state approved curriculum and ensure that those objectives are met when delivering this content.

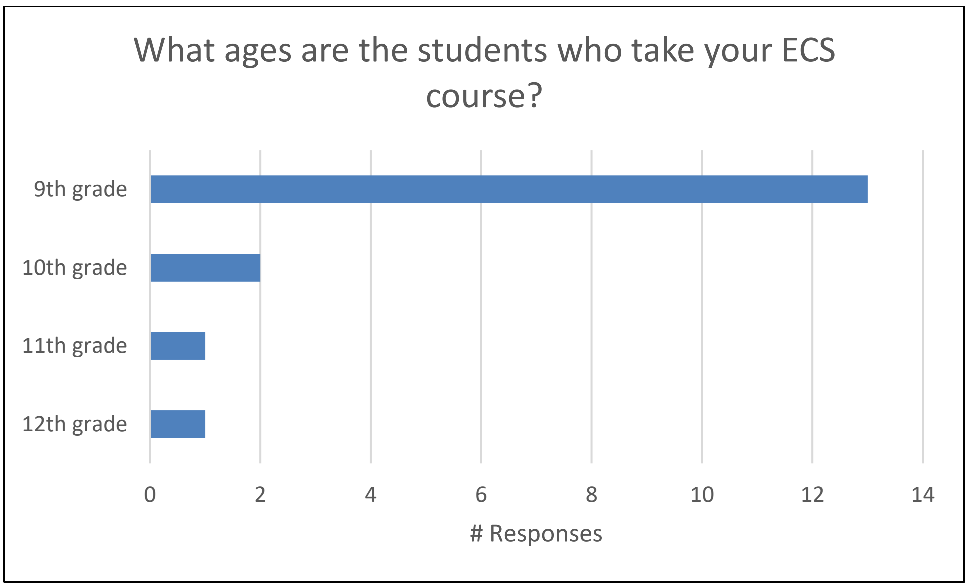
In our design meetings with Utah ECS I teachers, the consensus from the teaching community was that this lesson in the ECS I curriculum works great and did not need additional resources. What's working for you in your classroom? What kind of help would you like to see? Post a comment to discuss…

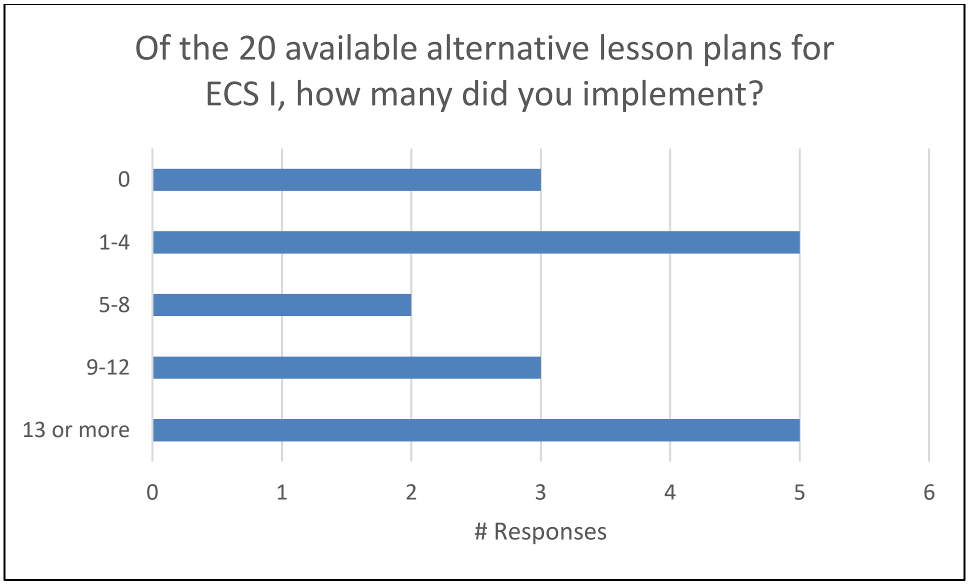
APPENDIX G – COMPLETE SURVEY RESULTS

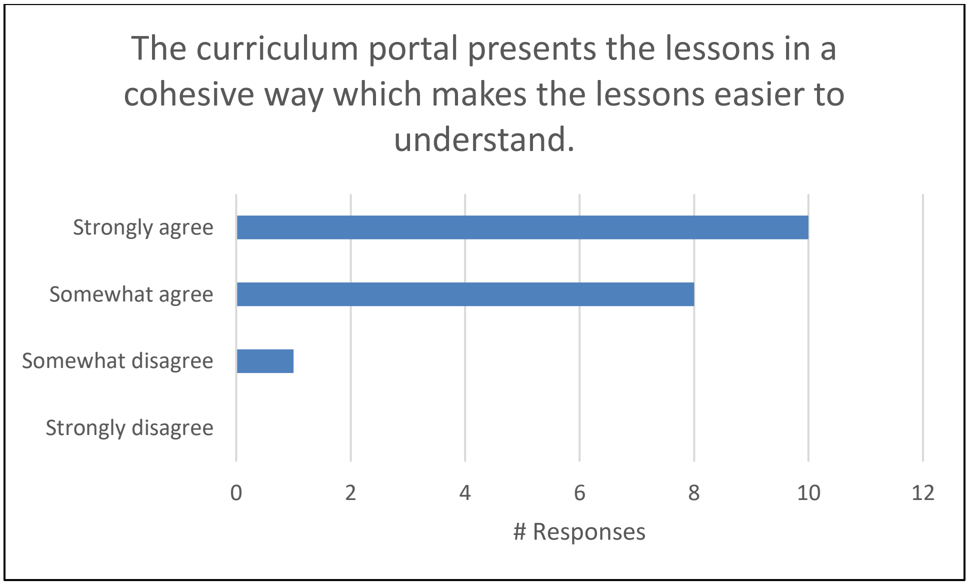


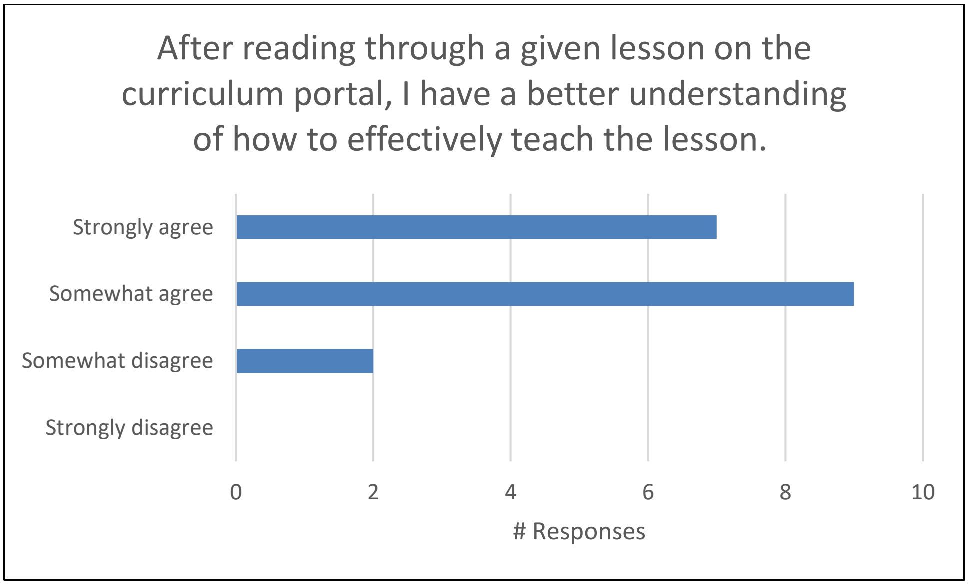


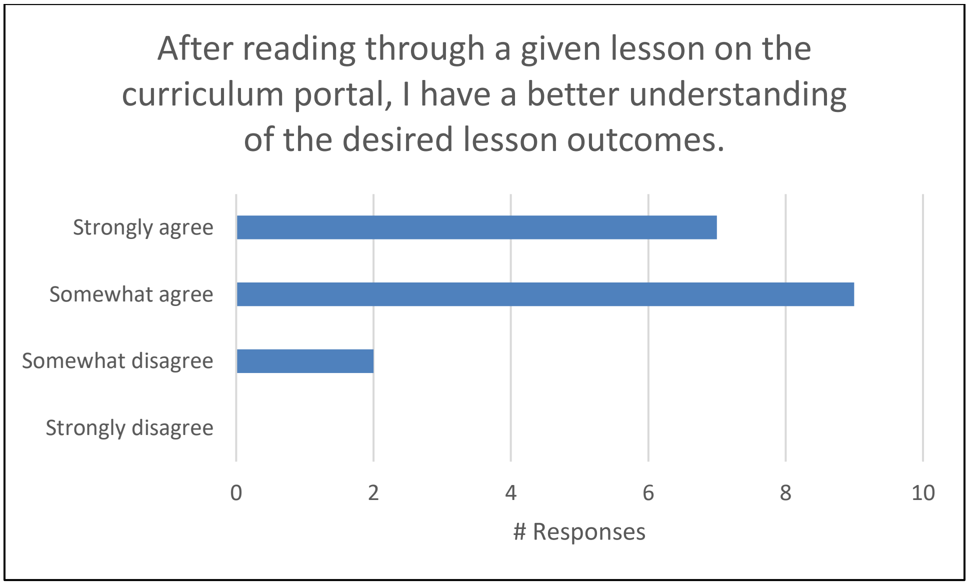

How long have you been using the ECS I course curriculum?
Over 6 months
4-6 months
2-4 months
Less than 2 months
0
2
4
6
8
10
12
14
Responses

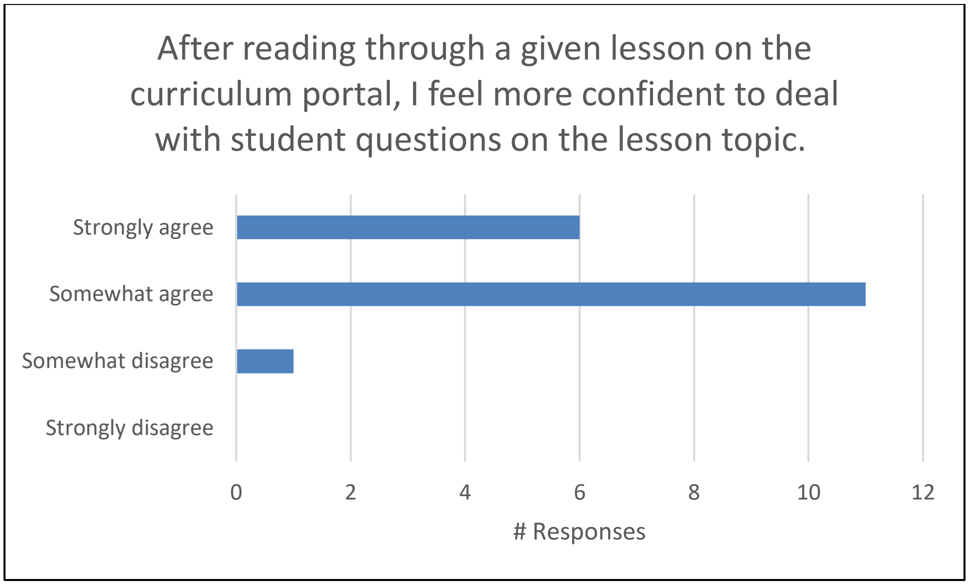

What ages are the students who take your ECS course?
9th grade
10th grade
11th grade
12th grade
0
2
4
6
8
10
12
14
Responses


Of the 20 available alternative lesson plans for ECS I, how many did you implement?
0
1-4
5-8
9-12
13 or more
0
1
2
3
4
5
6
Responses

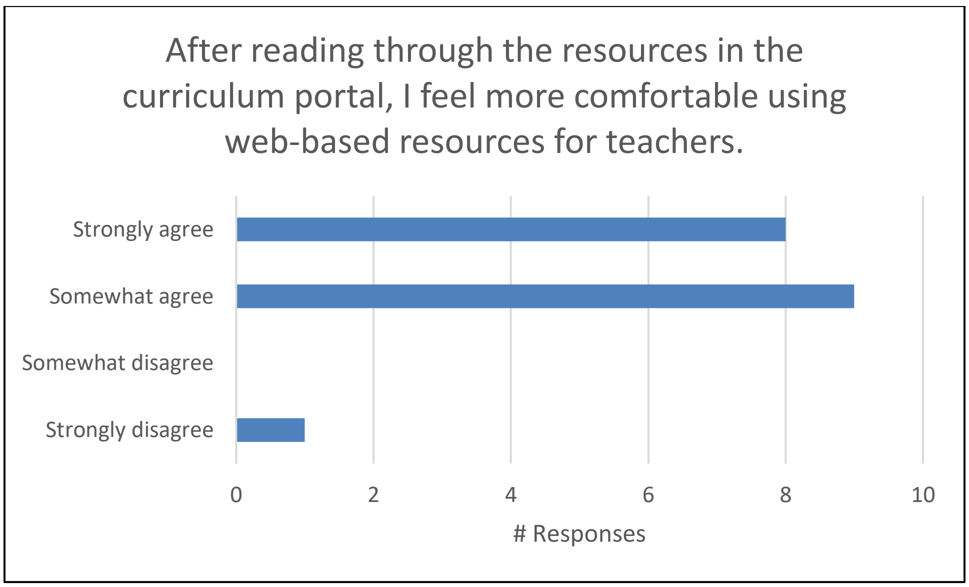

The curriculum portal presents the lessons in a cohesive way which makes the lessons easier to understand.
Strongly agree
Somewhat agree
Somewhat disagree
Strongly disagree
0
2
4
6
8
10
12
Responses


After reading through a given lesson on the curriculum portal, I have a better understanding of how to effectively teach the lesson.
Strongly agree
Somewhat agree
Somewhat disagree
Strongly disagree
0
2
4
6
8
10
Responses

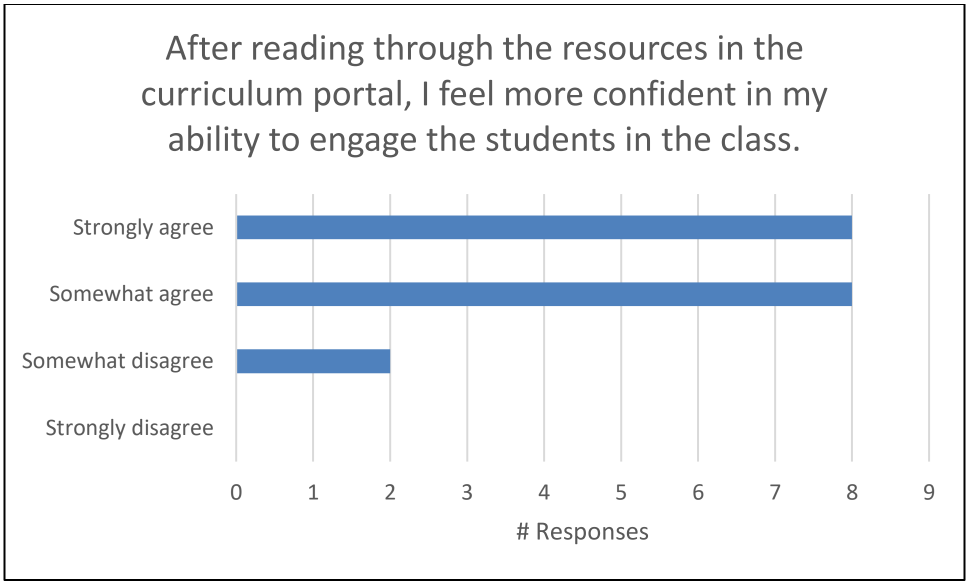

After reading through a given lesson on the curriculum portal, I have a better understanding of the desired lesson outcomes.
Strongly agree
Somewhat agree
Somewhat disagree
Strongly disagree
0
2
4
6
8
10
Responses

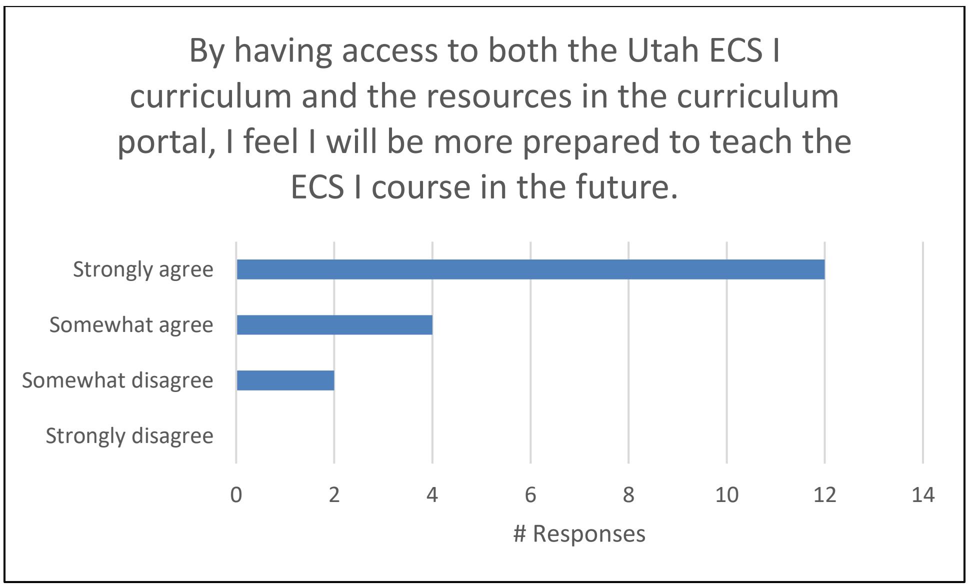


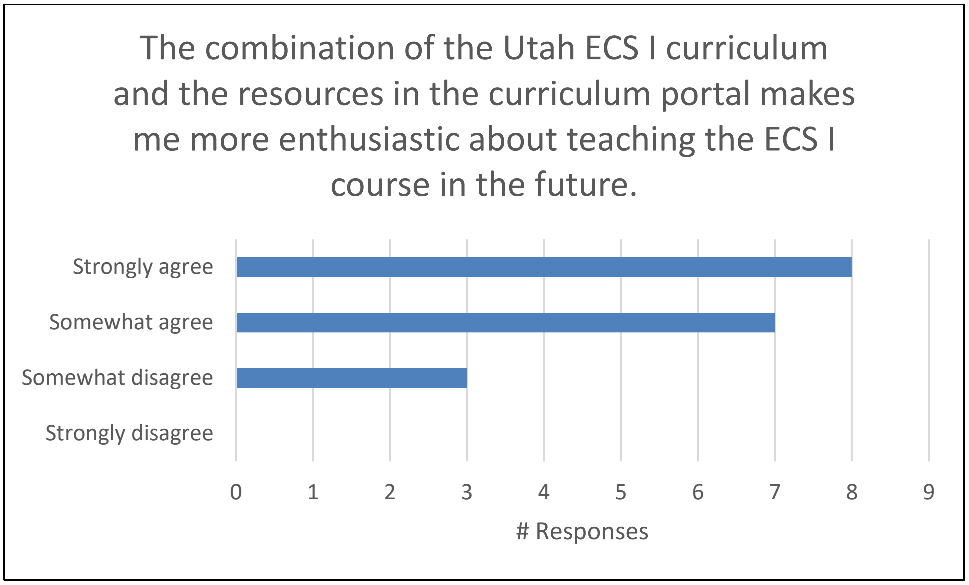
After reading through a given lesson on the curriculum portal, I feel more confident to deal with student questions on the lesson topic.
Strongly agree
Somewhat agree
Somewhat disagree
Strongly disagree
0
2
4
6
8
10
12
Responses

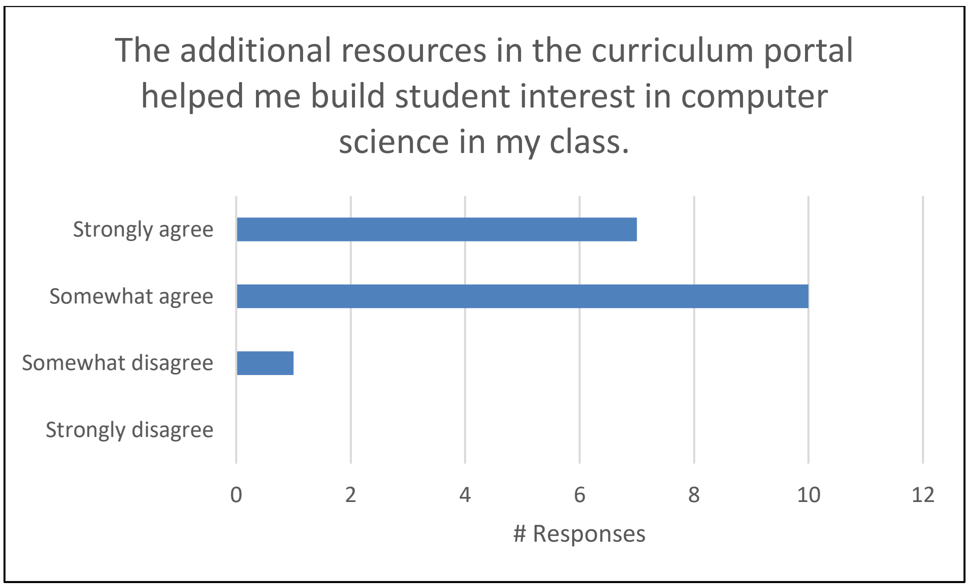


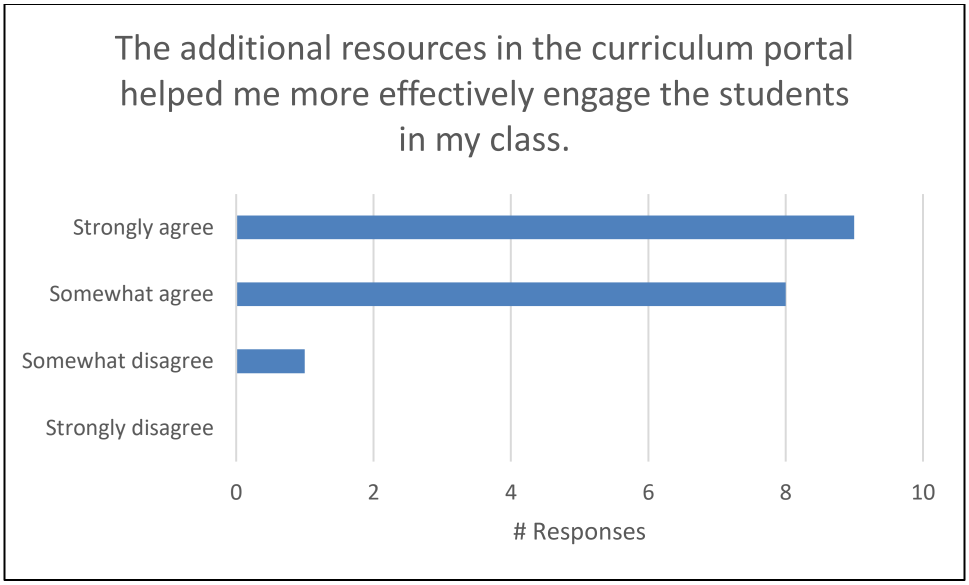
After reading through the resources in the curriculum portal, I feel more comfortable using web-based resources for teachers.
Strongly agree
Somewhat agree
Somewhat disagree
Strongly disagree
0
2
4
6
8
10
Responses

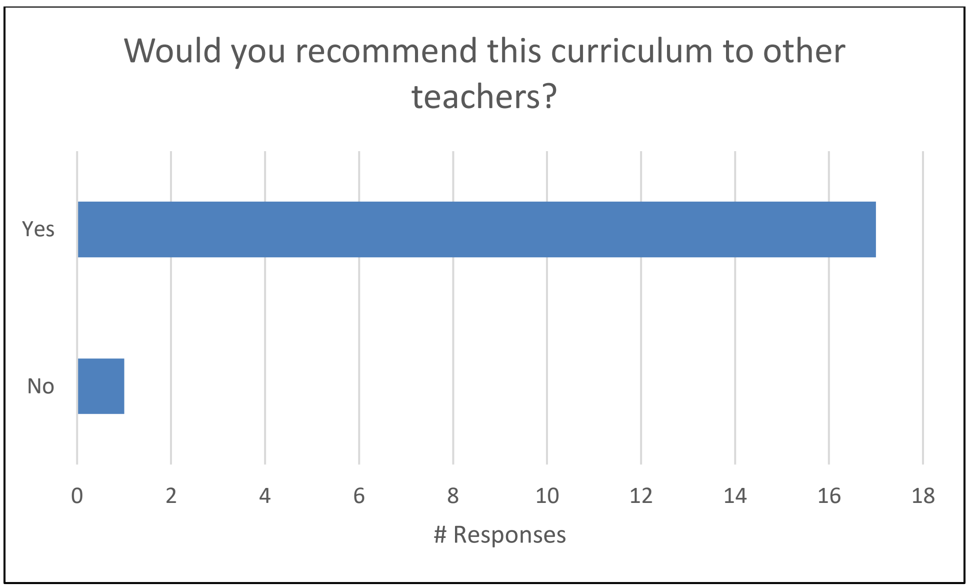

After reading through the resources in the curriculum portal, I feel more confident in my ability to engage the students in the class.
Strongly agree
Somewhat agree
Somewhat disagree
Strongly disagree
0
1
2
3
4
5
6
7
8
9
Responses


By having access to both the Utah ECS I curriculum and the resources in the curriculum portal, I feel I will be more prepared to teach the ECS I course in the future.
Strongly agree
Somewhat agree
Somewhat disagree
Strongly disagree
0
2
4
6
8
10
12
14
Responses


The combination of the Utah ECS I curriculum and the resources in the curriculum portal makes me more enthusiastic about teaching the ECS I course in the future.
Strongly agree
Somewhat agree
Somewhat disagree
Strongly disagree
0
1
2
3
4
5
6
7
8
9
Responses


The additional resources in the curriculum portal helped me build student interest in computer science in my class.
Strongly agree
Somewhat agree
Somewhat disagree
Strongly disagree
0
2
4
6
8
10
12
Responses


The additional resources in the curriculum portal helped me more effectively engage the students in my class.
Strongly agree
Somewhat agree
Somewhat disagree
Strongly disagree
0
2
4
6
8
10
Responses


Would you recommend this curriculum to other teachers?
Yes
No
0
2
4
6
8
10
12
14
16
18
Responses