# Programming as an Engineering Tool in K-12: e4usa+Programming. Introducing the Purple Thread

**Dr. Kenneth Reid, University of Indianapolis**

Kenneth Reid is the Associate Dean and Director of Engineering at the R. B. Annis School of Engineering at the University of Indianapolis. He and his coauthors were awarded the Wickenden award (Journal of Engineering Education, 2014) and Best Paper award, Educational Research and Methods Division (ASEE, 2014). He was awarded an IEEE-USA Professional Achievement Award (2013) for designing the B.S. degree in Engineering Education. He is a co-PI on the "Engineering for Us All" (e4usa) project to develop a high school engineering course "for all". He is active in engineering within K-12, (Technology Student Association Board of Directors) and has written multiple texts in Engineering, Mathematics and Digital Electronics. He earned a PhD in Engineering Education from Purdue University, is a Senior Member of IEEE, on the Board of Governors of the IEEE Education Society, and a Member of Tau Beta Pi.

**Dr. Cheryl Beauchamp, Regent University**

Cheryl Beauchamp serves as the Director for Regent University's Institute for Cybersecurity and chair of Regent's Engineering and Computer Science Department. She earned a Ph.D. in Engineering Education at Virginia Tech, a M.S. in Computer Science from George Mason University, and her M.Ed. from Regent University. Her research interests include computer science and cybersecurity education, student and faculty motivation in STEM education, teamwork, and online learning design.

**Dr. Stacy S Klein-Gardner, Engineering for US All**

Dr. Stacy Klein-Gardner serves as an Adjunct Professor of Biomedical Engineering at Vanderbilt University. She is the co-PI and co-Director of the NSF-funded Engineering For Us All (e4usa) project and now the Executive Director of e4usa, the non-profit company. Dr. Klein-Gardner is a Fellow of ASEE.

# Programming as an Engineering Tool in K-12: e4usa+Programming. Introducing the Purple Thread (Evaluation)


## Abstract

Nationwide, K-12 schools have implemented courses with the potential for AP® credit for Computer Science Principles, providing a broad overview of computing and computational thinking, exploring coding, learning how computing can affect the world around them, and obtaining college credit. Far fewer schools have implemented engineering in K-12, and those that have tend to be schools in more affluent districts. Committee reports from the National Academies from the early 2000s show that making a fundamental difference is difficult without specific educational standards and preservice teacher training, a situation that has shown little if any improvement.

The NSF-funded e4usa program was designed and implemented as an effort to provide engineering curriculum "for us all," with the intention of demystifying and democratizing engineering. The course authentically introduces "engineering," not with the intent to produce more engineering students, but to improve technological literacy and to allow students to discover their engineering identity, all while working through the engineering design processes to solve real problems. Students identify community-based problems and design engineering solutions while consulting with stakeholders, producing prototypes, and developing test plans. Course outcomes are clustered into color-coded tracks, including red (discover engineering), yellow (engineering in society), blue (engineering professional skills), and green (engineering design).

This paper focuses on the development of a course focused on programming in the e4usa sequence of courses. As the curriculum was more widely implemented, it became apparent that programming was a tool that added significant value when investigating solutions to problems using engineering design to address issues, and an issue of significant interest to K-12 schools. The course e4usa+Programming is in its pilot year.

This course uses MATLAB® as the programming language, since it is used prominently in engineering analysis and design. The course follows the path of AP CS Principles, focusing on the five big ideas: Creative Development, Data, Algorithms and Programming, Computer Systems & Networks, and Impact of Computing. The course also covers enough ground to allow students to sit for the MATLAB Associates Industry Recognized Credential (IRC). Development of the course required a new set of technical learning outcomes to the e4usa course sequence: the purple thread. Finally, the course requires students to use MATLAB in the solution of an authentic, community-focused engineering problem.

The paper will describe the course, the applicability of the MATLAB Associates IRC and AP CS Principles credit, tying the curriculum to specific state standards, and early results from a team of pilot teachers.

## Introduction

The growing demand for engineering professionals and calls for curricular reform in K-12 have led to an increase in the demand for and the number of pre-engineering programs. However, the integration of these programs into high school programs of study is not necessarily straightforward, as states' requirements and methods of implementation vary widely. Further exacerbating this issue is a lack of engineering experience among teachers, administrators, and legislators.

The National Academies launched a series of studies on integrating engineering into K-12 and found, among other reasons, difficulty due to lack of engineering within state standards and insufficient training for pre-service teachers [1-2]. Engineering involves open-ended, ill-defined problems, usually with multiple solutions [3-5], which is decidedly different from most high school curricula. In fact, introductory engineering courses and programs found at the university level are not standardized and often inconsistent, [6]; this does not portend well to high school programs.

The e4usa program, funded by the National Science Foundation, is a program designed to make engineering more democratized and demystified for all students (and teachers), not just those destined to study engineering in college. This innovative program has successfully been implemented in high schools throughout (and beyond) the United States [7-10].

Meanwhile, the demand for computer science in K-12 continues to grow as well. The AP® Computer Science Principles course [11], similar to the e4usa program, was designed to be accessible to students of all backgrounds, experience levels, and interests. Its overall framework is to introduce students to Computational Thinking practices with a focus on the impacts of computing and technology.

The e4usa team recognized the importance of including programming as a tool to develop engineering solutions and teamed with MathWorks to develop e4usa+Programming as a course. This course equips students with essential 21st-century programming knowledge and skills. Using MATLAB®, a powerful programming and computing platform utilized by millions of engineers and academic institutions worldwide, students explore algorithmic thinking and computational problem-solving through the lens of engineering design.

## Literature

Addressing engineering challenges in the 21st century involves large datasets and complex simulations, requiring computational thinking for engineers. Innovative solutions to these problems require problem-solving, data analysis, modeling and simulation, and efficiency. Computational Thinking (CT) provides a systematic approach to problem solving that can be more efficient and effective than traditional problem-solving approaches. The core elements of CT include Decomposition, Pattern Recognition, Abstraction, and Algorithm Design [12]. CT is not limited to computer scientists or programming, but recognizing its relevance for engineers, most college engineering education programs include computer application courses with a focus on modeling and simulation [13-14]. Undergraduate engineering programs include courses such as Computing Fundamentals at the University of Maryland, Computational Modeling of Engineering Systems at Arizona State University, Computing for Engineers and Duke University, and Introductory Programming for Engineers at Drexel University, to name a few. These courses provide students opportunities to program in higher level languages such as C++ and Python as well as work with industry relevant applications such as MATLAB and Computer Aided Design (CAD) software.

Although most undergraduate engineering programs include an introduction to programming or computing course, prior programming experience has been found to impact undergraduate student experience in an introductory level computer science course. Wilson and Shrock [15] found that previous computing experience, specifically formal class experience, had a positive influence on the course grade. Participants in the Diaz [16] study acknowledged that prior programming experience provided an advantage over those who did not have any prior exposure to computer programming. Some students questioned whether they were competent enough to be an engineer. Additionally, students with prior experience who openly displayed their superior knowledge exacerbated the anxiety felt by students who lacked a programming background. Engineering education programs are already challenged with student persistence as 40% of first year students do not make it through their first year [17]. Although not as high, a 2019 study reported a 28% failure rate of students in Introduction Programming courses [18].

Recognizing the need for CT skills and the positive impact prior programming experience plays in engineering education persistence, e4usa integrates CT and programming skills in its curricula to equip high school students with the necessary skills to provide innovative solutions that address challenges in our modern technological world.

## Background e4usa Curriculum

The e4usa course is unlike most, if not all, existing high school engineering curricula. The intent is not necessarily to produce more engineers or introduce specific engineering skills, but to introduce creative challenges where engineering is required to develop a solution to a given problem. The curriculum’s focus is on engineering-centric skills (e.g. teamwork,

interdisciplinary thinking, critical thinking) rather than specific, discipline-focused engineering content. The course learning outcomes were initially developed through a series of workshops engaging experts in K-12 engineering, beginning in December 2018 with over 100 attendees, from universities, national research organizations, K-12, and other experts. The initial design was framed around the First Year Engineering Classification Scheme [19], developed to classify all content found in first-year, multidisciplinary Introduction to Engineering courses at the university level, and El Sawi's curricular development framework [20].

From here, the introductory course curriculum was developed around the following threads (Table 1):

**Table 1: Course Outcomes for each thread**

| | |
|---|---|
| | **Red Thread: Discover Engineering**<br>● Iterate and evolve the definition of what it means to engineer and be an engineer.<br>● Recognize the value of engineering for all regardless of one's potential career.<br>● Explain and apply ethical & societal considerations when exploring an engineering problem. |
| | **Yellow Thread: Engineering in Society**<br>● Explore the impacts of past engineering successes and failures on society as a whole.<br>● Recognize and investigate the world's greatest challenges and the role that engineering plays in solving these challenges (e.g., Engineering Grand Challenges, UN sustainability goals, etc.).<br>● Integrate cross-disciplinary thinking and expertise to inform design solutions that add value to society.<br>● Identify and analyze issues when bringing a solution to scale. |
| | **Blue Thread: Engineering Professional Skills**<br>● Use various engineering communication methods.<br>● Collaborate effectively in a team.<br>● Develop, implement, and adapt a project management plan. |
| | **Green Thread: Engineering Design**<br>● Identify and describe a problem that can be solved with a potentially new product or process.<br>● Identify appropriate stakeholders and content experts and evaluate their input.<br>● Plan and conduct research by gathering relevant and credible data, facts, and information.<br>● Articulate appropriate STEM practices and principles in the design<br>● Evaluate solution alternatives and select a final design by considering assumptions, tradeoffs, criteria, and constraints.<br>● Create a prototype.<br>● Create and implement a testing plan to evaluate the performance of design solutions.<br>● Apply iteration to improve engineering designs.<br>● Articulate and reflect on how an engineering design process could be applied to solving a problem |

Note that there is no significance to the colors of the threads, but they serve to distinguish and categorize outcomes to the thread. Students progress through a project-based curriculum, exploring 'engineering' as a problem-solving discipline, exploring their engineering identity, and discovering the effect of ethics, society, and other constraints on engineering solutions. The e4usa program has students follow an engineering design process multiple times, solving problems at personal, local, and more global levels, while interacting with stakeholders. The curriculum stresses problem-solving, design thinking, creativity, innovation, and collaboration.

The course is meant to introduce engineering to everyone, not just students with an engineering background or understanding, and certainly not only students who have prepared to enter engineering. For example, the math prerequisite for the e4usa course is algebra 1. The course is not intended to be a skills-based course: in fact, specific skills such as programming or CAD are not required (but may be used if applicable) [9,10]

The e4usa curriculum is housed on teachengineering.org [21], giving teachers a portal to access the curriculum and submit feedback to the curriculum developers.

The curriculum is organized hierarchically with a Unit page for each of the eight units (including two parallel paths through Unit 2 where the Engineering Design process is introduced, allowing teachers to focus on a water filter or a wind energy project), with a Lesson page for each lesson within a unit and Activity pages for each activity embedded within each lesson. Additional teacher-facing content is available through links, including slide decks and student handouts.

As the curriculum has matured, it became clear that teachers had difficulty progressing through eight units, including three distinct engineering designs. The original e4usa course has evolved to a two-course sequence, allowing more time throughout each step of the engineering design process. The course has been reconfigured into two courses, named e4usa+making and e4usa+design. An accelerated version of the course with older students who typically have more communication and teamwork skills already developed remains in the e4usa Legacy course.

Each lesson contains Course Objectives (COs) and Student Learning Outcomes (SLOs), housed under the Learning Outcomes heading in teachengineering. COs and SLOs are organized based on four encompassing learning outcomes called threads: Engineering Professional Skills (Blue thread), Connect with Engineering (Red thread), Engineering in Society (Yellow thread), and Engineering Design (Green thread). COs and SLOs have been adjusted and changed overtime as the e4usa curriculum has developed, focusing on the key components of engineering in the high school classroom. Currently, there are nineteen COs. Each of our 57 SLOs is assigned a CO, and both SLOs and COs are used to guide teachers and students through the expectations and skills of each unit and lesson.

With the COs and SLOs are rubrics that serve as assessments. Rubrics for each are available on Teachengineering.org under the Assessments heading of each lesson. Teachers use the rubrics to

assess how each student performed on a SLO, measuring from the scores “Not Submitted” (0); “Novice” (1); “Developing” (2); and “Proficient” (3). Each score is described to assist teachers in the criteria of what score a student will receive based on their performance. Rubrics have also been updated with more specific language and criteria based on teacher feedback.

**Professional Learning**

The e4usa program has a goal to demystify engineering, which applies to students and teachers. Teachers in the program have ranged from those with prior engineering experience to music and history teachers. A professional development framework onboards and supports new and existing teachers with annual summer and winter training sessions and the establishment of communities of practice. The professional development process is endorsed by the American Society of Engineering Education. Endorsement ensures that e4usa’s professional learning is properly designed to emphasize the aspects of engineering content, pedagogy, curriculum, and assessment that e4usa thinks is essential and documents that.

e4usa provides hands-on, collaborative learning experiences that prepare teachers to successfully support their students in using the e4usa curriculum, developing technical knowledge and skills, and creatively solving realistic problems with their peers. All of e4usa’s professional learning options ensure that teachers are well-prepared and supported throughout the program.

**Teachers, Students, Impact**

Since its inception in the 2019-2020 academic year, e4usa has seen over 7700 students and 100 teachers complete the program. Student projects have impacted the Nashville Zoo, communities local to individual schools, and multiple systems within schools. Twenty-five colleges and universities in 14 states and 2 countries award credit for students who complete the program [22].

**Computer Science and Programming in K-12**

Demand for computing education in K-12 has experienced significant growth in the past few decades with many governments across the world requiring the integration of computing courses. In 1983, the National Commission on Excellence in Education, a group convened under President Ronald Reagan’s White House recommended a computer science course be included as part of high school graduation requirements [23]. In 2016, President Barack Obama’s Computer Science (CS) For All Initiative recognized that other nations were making strides in CS education while only a quarter of the schools in the United States offered programming and coding. His CS for All initiative supported the development of teaching and learning materials, professional development for educators, the Advanced Placement CS Principles framework, and continued research on approaches and impacts [24] A 2019 literature review on introductory programming from 2003 to 2017 found many initiatives were driven by government mandates to

teach programming; however, educator resources and training are still lacking [25]. Additionally, research regarding effective computer education pedagogical approaches is still needed.

**Programming in K-12 standards**

Although only six states had adopted K-12 CS standards in 2017 after President Obama's CS For All initiative launch, 44 states, as of 2024 have CS standards. Additionally, the number of high schools that have CS classes available has nearly doubled from 2017-2018 [26]. Although standards may vary from state to state, many are informed by the CS Standards that were developed by the Computer Science Teachers Association (CSTA). The CSTA standards, which are currently being revised with a scheduled 2026 release, were developed from the CSTA K-12 CS Framework which provides grade level guidance from which the standards then detail measurable student performance expectations. The Framework includes Core Concepts and Practices as seen in Table 2 [27].

**Table 2: Concepts and Practices of the Computer Science Framework for K-12**

| The Concepts and Practices of the Computer Science Framework for K-12 | |
|---|---|
| Core Concepts | Core Practices |
| Computing Systems | Fostering a Computing Culture for all |
| | Collaborating Around Computing |
| Networks and the Internet | Recognizing and Defining Computational Problems |
| Data and Analysis | Developing and Using Abstractions |
| Algorithms and Programming | Creating Computational Artifacts |
| Impacts of Computing | Testing and Refining Computational Artifacts |
| | Communicating About Computing |

**Advanced Placement Computer Science Principles:** The College Board, with the support of the NSF, developed the Advanced Placement Computer Science Principles (AP CSP) course to provide computing coursework that was considered more accessible and engaging to all students. The course was designed with Computer Science For All in mind, recognizing the need for technology innovations to reflect the innovative workforce that it supports. Part of the goal when creating the course was to focus on computational thinking practices that would ensure students would be active creators of technology instead of passive users of it [28]. The main concepts of

the course are called the Big Ideas. These Big Ideas and their subtopics [11], as seen in Table 3, are similar to the CSTA Core Concepts and Practices.

**Table 3: AP Computer Science Principles Big Ideas**

| AP Computer Science Principles Big Ideas |
|---|
| Creative Development |
| Data |
| Algorithms and Programming |
| Computer Systems and Networks |
| Impact of Computing |

The course was launched in the fall of 2016, with 44,330 students taking the AP CSP exam in 2017. By 2024, 175,000 students had taken the exam [29,30]. These growing trends also support the Computer Science for All initiative as prior research indicates a strong link between students taking the AP Exam and majoring in STEM fields later in college [31]. A 2020 report also found that AP CS Principles served as a STEM field entryway for first-generation students [32].

**Industry Recognized Credentials in Career and Technical Education Programs**

In addition to college preparation courses, career and technical education (CTE) programs also recognize the importance of computing knowledge and skills. According to the National Center for Education Statistics, CTE includes both high school level courses and postsecondary level programs that focus on skills and knowledge that are required for a specific job or field of work [33]. As the cost of higher education continues to escalate and students question the value of college, CTE programs are responding by incorporating industry-recognized credentials (IRCs) into their programs [34]. Benefits of an IRC for students may include increased employment prospects as well as validation of real-world skills. Several states, including Virginia, Texas, and Ohio, are incorporating IRCs in their CTE curriculum as a requirement for course credit or program completion [35]. The Blueprint for Maryland's Future [36] includes a requirement for 45% of students to receive at least one IRC by the end of 2030.

**+Programming: Course Description and Outline**

With a strong emphasis on programming in K-12, and its wide adoption in state educational standards, the course e4usa+Programming was conceived. This course aims to align with Computer Science learning standards developed by the Computer Science Teacher Association

(CSTA), which are widely adopted in career and technical education programs across the country.

The +Programming course is designed to meet the requirements for the AP Computer Science Principles (AP CSP) course. The five units of the course align with the five Big Ideas, its subtopics, and skills. Using a reverse engineering approach, the course developers designed the +Programming course using the AP CSP course requirements as its framework. This intentionally ensured the curriculum aligned with the learning objectives for AP CSP and thus the CSTA standards for Computer Science as well. When schools offer AP courses they must obtain College Board approval for their curriculum. This is referred to as their AP Course Audit process. Ensuring the units clearly aligned with the Big Ideas of AP CSP ensured that educators would be able to use the +Programming course as the College Board approved curriculum for AP CSP. Students complete a Create Performance Task in advance of the exam, allowing students to have the time necessary to do real coding. A distinguishing feature of +Programming compared to other College Board approved curricula for AP CSP is in this Creative Performance Task (CPT) component of the course. The CPT is part of the AP exam and consists of an extensive program and supporting material developed by the student. For students in +Programming, they are required to collaborate with others and ensure their project contributes to a solution for an engineering problem for an authentic client, ensuring that the coding is not merely just an exercise. e4usa aims for students to see computer programming as one of many tools that engineers use to solve additional types of engineering problems or to solve engineering problems more effectively or more efficiently.

Additionally, the +Programming course includes material necessary for students to pass the MathWorks Associates Exam. Access to the exam comes with access to a thorough set of training modules posted by MathWorks. These materials may be accessed while the student teams are solving their authentic engineering problem as needed or they may come after the completion of the CPT as part of the AP CSP course requirements.

The +Programming course maintains the strategic focus of e4usa including the requirement of only Algebra as a prerequisite, an appeal to all students, and an assumption that incoming students have no prior programming experience. This course does have new, more technical outcomes necessary to address both the AP Computer Science Principles and the MathWorks Industry Recognized Credentials (Associate level).

**Introducing: The Purple Thread**

The +Programming course is being piloted in the 2024-2025 academic year at four schools: two in Maryland, one in North Carolina and one in New Mexico. Two teachers have fairly extensive programming experience, while one had almost no experience. All four teachers have taught the

e4usa course for multiple years and have experience with leading teams in all phases of engineering design. A total of 55 students are enrolled.

The +Programming course is structured by unit, largely following the structure of the AP Computer Science Principles course:

**Unit 1: Introduction to Computer Science and MATLAB**

Students explore computing using MATLAB, and its use as in manipulating and exploring data. MATLAB applications are explored. This unit introduces computer science concepts and has students run some MATLAB livescripts, showing the power and potential of developing programs. Students are also walked through the MATLAB Onramp, introducing students to the operating environment and computer programming basics.

**Unit 2: Data**

Students explore the structure and nature of data, including number systems commonly used in computer science, and the use of MATLAB for data analysis. Students explore data formats, storage of data, and manipulating and analyzing big data.

**Unit 3: Algorithm and Programming**

Students develop algorithms and implement algorithms with MATLAB. This unit has almost all of the instruction on computer programming, taking students from a more formal introduction of programming and MATLAB through development of programs. This unit ends with demonstrations of mastery including the Create Performance Task (CPT), especially applicable to students taking the course for AP credit and MATLAB Industry Recognized Credentials, for students interested in sitting for the MATLAB Associates Exam.

**Unit 4: Computer Systems and Networks**

Students explore networking basics, including the structure and operation of the Internet, fault tolerance, and parallel and distributed computing.

**Unit 5: Impacts of Computing**

Computational thinking and problem-solving using computing are featured. Students also explore the societal and ethical considerations of the use of computers and computing

more deeply. Collaborative computing, privacy and security, and historical examples are featured.

The creation of a new course and its tie to the AP CS Principles course meant that additional learning outcomes were needed. A new, color-coded thread was added (Table 4), based on the Computational Thinking Practices listed in the AP CS Principles course:

**Table 4: Purple Thread**

| **Purple Thread: Engineering Connection** |
| --- |
| ● Demonstrate an understanding of computing systems<br>● Write programs using the knowledge of and skills related to algorithms and programming.<br>● Demonstrate knowledge of and skills related to data analysis.<br>● Report on the impacts of computing.<br>● Demonstrate an understanding of the networks and the internet.<br>● Effectively collaborate in computing.<br>● Demonstrate understanding of the impacts of computing. |

The course is in its pilot year, and the development of +Programming continues, largely following the success of the original e4usa curriculum, with iterative curriculum development incorporating teacher input [37].

## Conclusion

The demand for engineering and computer programming in K-12 continues to grow, and impediments to implementation remain in place, including lack of preparation for incoming teachers and inconsistent approaches in state-level standards. e4usa courses have proven successful in introducing engineering to high-school students nationwide, and professional learning strategies have been successful in onboarding teachers regardless of their background. The new +Programming course introduces programming using MATLAB. Students will use MATLAB to develop programs that are part of an engineering solution to a problem, thus understanding programming as a useful tool and developing an appreciation for authentic programming.

**References**

[1] National Academy of Engineering and National Research Council. 2009. Engineering in K-12 Education: Understanding the Status and Improving the Prospects. Washington, DC: The National Academies Press. https://doi.org/10.17226/12635.

[2] National Academy of Sciences, National Academy of Engineering, and Institute of Medicine. 2007. Rising Above the Gathering Storm: Energizing and Employing America for a Brighter Economic Future. Washington, DC: The National Academies Press. https://doi.org/10.17226/11463.

[3] S. Brophy, S. Klein, M. Portsmore, & C. Rogers, "Advancing engineering education in P-12 classrooms," *Journal of Engineering Education*, 97(3), 369-387, 2008.

[4] National Research Council. *Standards for K-12 Engineering Education?* Washington, DC: The National Academies Press. 2010. https://doi.org/10.17226/12990.

[5] National Academies of Sciences, Engineering, and Medicine. *Building Capacity for Teaching Engineering in K-12 Education*. Washington, DC: The National Academies Press. 2020. https://doi.org/10.17226/25612.

[6] K. Reid, D. Reeping, E. Spingola, "A taxonomy for introduction to engineering courses." International Journal of Engineering Education, 35(1), 2-19, 2018.

[7] B. Berhane, M. Dalal, S. Klein-Gardner, A. Carberry, K. Reid, C. Beauchamp, J. Kouo, & D. Pines. *Understanding the 'us all' in Engineering 4 Us All through the experiences of high school teachers.* CoNECD Virtual Conference, 2021.

[8] R. Figard, M. Dalal, J. Roarty, S. Nieto, & A. Carberry, *Understanding High School Student Experiences in an Engineering Course Designed For All*, American Society for Engineering Education (ASEE) Conference & Exposition, Minneapolis, MN, 2022.

[9] K. Reid, T. Greisinger, *A high school engineering curriculum focusing on discovery, design, professional skills and society*. World Engineering Education Forum/Global Engineering Deans Council (WEEF/GEDC), 2021.

[10] K. Reid, K. Ladeji-Osias, C. Beauchamp, M. Dalal, T. Griesinger, & E. Eagle, *Design by thread: The e4usa curriculum*. ASEE/IEEE Frontiers in Education Conference (FIE), 2020.

[11] AP Computer Science Principles, URL: https://apcentral.collegeboard.org/courses/ap-computer-science-principles, accessed 1/13/2025.

[12] J.M. Wing, Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717-3725., 2008. https://doi.org/10.1098/rsta.2008.0118

[13] H.A. Diefes-Dux, T. Moore, J. Zawojewski, P.K. Imbrie, & D. Follman, *A framework for posing open-ended engineering problems: Model-eliciting activities*, Proceedings of the 34th ASEE/IEEE Frontiers in Education Conference, Savannah, GA, 2004. https://doi.org/10.1109/FIE.2004.1408556

[14] N. Zhang, G. Biswas, K.W. McElhaney, S. Basu, E. McBride, & J.L. Chiu, *Studying the interactions between science, engineering, and computational thinking in a learning-by-modeling environment*. In: I. Bittencourt, M. Cukurova, K. Muldner, R. Luckin, & E. Millán, (eds) Artificial Intelligence in Education. AIED 2020. Lecture Notes in Computer Science, vol 12163. Springer, Cham. https://doi.org/10.1007/978-3-030-52237-7_48

[15] B.C. Wilson & S. Shrock, *Contributing to success in an introductory computer science course: a study of twelve factors*. ACM SIGCSE Bulletin, 33(1), 184-188, 2001. https://doi.org/10.1145/366413.364581

[16] N.V. Mendoza Diaz, D.A. Trytten, R. Meier, & J.M. Moore, *The Impact of Prior Programming Experience on Computational Thinking in First-Year Engineering Experience.* Proceedings of the CoNECD Conference, 2022.

[17] P.C. Boylan-Ashraf & J.R. Haughery, *Failure rates in engineering: Does it have to do with class size?* Proceedings of the 2018 ASEE Annual Conference & Exposition. https://doi.org/10.18260/1-2--30515.

[18] J. Bennedsen, & M.E. Caspersen, *Failure rates in introductory programming: 12 years later*. ACM Inroads, 10(2), 30–36, 2019. https://doi.org/10.1145/3324888

[19] K. Reid, D. Reeping, E. Spingola, *A taxonomy for introduction to engineering courses.* International Journal of Engineering Education, 35(1), 2-19, 2018.

[20] G. El Sawi, PhD, Curriculum Development Guide: Population Education for Non-Formal Education Programs of Out-of-School Rural Youth, Food and Agriculture Organization of the United Nations, URL: http://www.fao.org/3/ah650e/AH650E00.htm, accessed 5/20/2020.

[21] teachengineering website: [URL: https://www.teachengineering.org/e4usa], accessed 2/1/2025.

[22] e4usa website: [URL: https://e4usa.org/], accessed 1/1/2025.

[23] National Commission for Excellence in Education, *A Nation at Risk: The Imperative for Education Reform*. The Elementary School Journal, 84(2), 112-130, 1983.

[24] National Science Foundation, *Building a foundation for CS for All*. (n.d.). URL: https://www.nsf.gov/news/news_summ.jsp?cntn_id=137529&org=NSF&from=news, accessed 1/10/2025

[25] C. Szabo, J. Sheard, A. Luxton-Reilly, Simon, B.A. Becker, & L. Ott, *Fifteen years of introductory programming in schools: a global overview of K-12 initiatives.* Proceedings of the 19th Koli Calling International Conference on Computing Education Research (pp. 1-9), 2019. https://doi.org/10.1145/3364510.336451

[26] B. Twarek, *What is the State of CS Education in 2024?* Computer Science Teachers Association, October 24, 2024 https://csteachers.org/what-is-the-state-of-cs-education-in-2024/, accessed 1/1/2025.

[27] *K–12 Computer Science Framework*. (n.d.). K–12 Computer Science Framework. k12cs.org. URL: https://k12cs.org/, accessed 1/1/2025.

[28] National Science Foundation, *College Board launches new AP Computer Science Principles course*. (n.d.). URL: https://www.nsf.gov/news/news_summ.jsp?cntn_id=133571#:~:text=To%20address%20the%20challenge%20of,(AP)%20Computer%20Science%20Principles, accessed 1/1/2025.

[29] *AP Computer Science Principles Exam Score Distributions – AP Students*. (n.d.). URL: https://apstudents.collegeboard.org/about-ap-scores/score-distributions/ap-computer-science-principles. Accessed: 1/1/2025

[30] C. Avery, O. Gurantz, M. Hurwitz & J. Smith, *Shifting College Majors in Response to Advanced Placement Exam Scores,* SSRN, March 2017. http://dx.doi.org/10.2139/ssrn.2869233.

[31] K.D. Mattern, E.J. Shaw, M. Ewing & College Board, *Advanced Placement exam participation is AP exam participation and performance related to choice of college major*? In College Board Research Reports (Research Report No. 2011–6). College Board. 2011. URL: https://files.eric.ed.gov/fulltext/ED561044.pdf, accessed 1/1/2025.

[32] J. Wyatt, J. Feng, M. Ewing & College Board. (2020). *AP Computer Science Principles and the STEM and Computer Science Pipelines*. 2020. URL: https://apcentral.collegeboard.org/media/pdf/ap-csp-and-stem-cs-pipelines.pdf. Accessed 1/1/2025.

[33] NCES, *COE - Career and Technical Education in the United States*. (n.d.). URL: https://nces.ed.gov/programs/coe/indicator/tob#:~:text=Career%20and%20technical%20education%20(CTE)%20is%20defined%20as%20courses%20(,jobs%20or%20fields%20of%20work., accessed: 1/10/2025

[34] M. Giani, A.M. Northern, M.J. Petrilli, Thomas B. Fordham Institute, Barr Foundation, Thomas B. Fordham Foundation, S. Dougherty, D. Fowler, K. Johnson, A. Tyner, E.P. Finn, Jr., V. McDougald, W. Rost, P. Tatz, D. Williams, L. Brennan & L. Gueguen, *How attaining Industry-Recognized credentials in high school shapes education and employment outcomes.* Thomas B. Fordham Institute, 2022. URL: https://fordhaminstitute.org/sites/default/files/publication/pdfs/final-industry-recognized-credentials-08232022irc.pdf, accessed 1/10/2025.

[35] The Thomas B. Fordham Institute, *Industry-recognized credentials are helpful but not transformative for high school students*. (n.d.). URL: https://fordhaminstitute.org/national/commentary/industry-recognized-credentials-are-helpful-not-transformative-high-school. Accessed: 1/1/2025.

[36] Maryland Public Schools, *Blueprint - Maryland State Department of Education, URL:* https://blueprint.marylandpublicschools.org/, accessed 1/10/2025

[37] M. Dalal, K. McKeown, K. Reid, V. Vinayak, *Evolving Together: The Role of Teacher Feedback in Shaping the e4usa Curriculum (Evaluation)*, Proceedings of the 2025 ASEE Annual Conference & Exposition.