Advanced Computing Through Short-Format Immersive Camps: Cyberforensics, AI, and Python for Novice Learners*

Johnathan Yerby[1] and Mehakpreet Kaur[2]
Department of Computer Science
Mercer University
Macon, GA 31207
[1]yerby_jm@mercer.edu
[2]kaur_m@mercer.edu

Abstract

This case study examines a university-led initiative that delivered intensive, hands-on computing camps for high school students in Middle Georgia, a mid-sized region with limited access to advanced computing instruction in K–12 settings. The program included three distinct four-day camps focused on cyberforensics, artificial intelligence, and Python programming. Students ages 13 to 18 engaged with real-world tools, gamified challenges, and scaffolded learning experiences. Designed for participants with little or no prior exposure, each camp emphasized ethical reflection, scenario-based problem solving, and AI-assisted development. Participants reported increased technical confidence and greater interest in computing careers. While the short-format model required individualized support, it successfully introduced complex topics in an approachable format. These results suggest that targeted outreach programs can help expand access to advanced computer science content in regions where such opportunities remain limited.

1 Introduction

National efforts to expand K–12 computer science education have intensified over the past decade, with growing emphasis on specialized areas such as cybersecurity, artificial intelligence (AI), and programming. Yet access to rigorous, hands-on computing experiences remains largely concentrated in urban and high-resource environments. In Middle Georgia, a region that is not fully rural or fully urban, high school students have limited opportunities to engage meaningfully with advanced computing topics. Most local programs focus on general STEM exposure or basic coding, leaving a gap for students who are ready to explore more challenging, career-relevant content.

This lack of early exposure can carry significant long-term consequences. Students without access to advanced technical learning opportunities are less likely to pursue computer science majors in college, qualify for competitive scholarships or internships, or develop the portfolio-ready skills increasingly sought by employers. In under-resourced regions like Middle Georgia, this results in a growing opportunity gap, both in postsecondary education and in the broader technology workforce.

To address this need, a university-led initiative offered three four-day computing camps in cyberforensics, artificial intelligence, and Python programming. Designed to be rigorous, hands-on, and accessible regardless of prior experience, the camps emphasized real-world tools, gamified learning, and scenario-based problem solving. Each session served 8 to 15 students for four hours per day during summer 2025.

2 Problem

Despite national initiatives to expand computing education, many mid-sized and under-resourced regions still lack sustained access to advanced technical instruction. By focusing on a mid-sized, under-resourced region, this work extends research beyond the rural or urban dichotomy that dominates CS outreach literature. In Middle Georgia, most high school computing opportunities focus on digital literacy or basic programming, without progression to applied or high-level content. As a result, motivated students often lack the means to develop deeper skills or explore academic and career pathways in computing. There is a clear need for engaging and scalable models that introduce advanced topics in an accessible way for students with little to no prior experience.

3 Literature Review

Early exposure to computer science (CS) and cybersecurity significantly influences students' academic and career pathways [19]. At the high school level, effective computing education is increasingly recognized as essential for workforce readiness and for navigating a world shaped by data, automation, and artificial intelligence. However, access to high-quality instruction varies widely by geography and socioeconomic status. This review synthesizes research across four areas: early exposure to computing, regional disparities, the role of informal outreach, and instructional strategies for engaging novice learners.

Studies consistently show that early exposure fosters sustained interest and persistence in computing fields ([8], [19]). Hands-on, project-based, and gamified environments outperform traditional lectures in promoting engagement and retention, especially in pre-college settings ([6], [10], [14]). Yet, most outreach programs target urban or affluent communities, limiting access for students in economically constrained or mid-sized regions.

There is growing urgency to equip students with artificial intelligence (AI) and machine learning (ML) literacy, not just as tool users but as future contributors to the AI-driven economy [17]. The AI4K12 Initiative outlines five "big ideas" that K-12 students should understand about AI: perception, representation and reasoning, learning, natural interaction, and societal impact. These benchmarks provide a clear, age-appropriate framework for integrating AI education across grade levels. Programs such as Code.org's AI curriculum and the AI4K12 guidelines demonstrate that advanced topics like AI/ML, cybersecurity, and data science can be introduced successfully at the secondary level when supported by effective scaffolding and accessible tools. However, many current outreach efforts remain limited in scope and fail to align explicitly with these standards. Topics such as cyberforensics are still rarely included in K-12 curricula, despite their increasing relevance.

3.1 Resource-Limited Access

Persistent disparities in computing education access are well documented. Urban and affluent schools are far more likely to offer Advanced Placement CS courses, specialized electives, and extracurricular enrichment than their rural or mid-sized counterparts [2]. Even in states with strong CS mandates, implementation gaps persist in under-resourced districts, where offerings often remain limited to digital literacy. Middle Georgia exemplifies this divide: not rural but under-resourced, it lacks the infrastructure and academic computing programs available in larger metropolitan areas. Many capable students lack a pathway to explore computing before college.

3.2 Effectiveness of Summer Camps

Summer computing camps offer a flexible model for technical enrichment outside traditional classrooms. Research shows these informal settings can build student confidence, foster belonging, and provide access to real-world tools and problem-solving scenarios ([1], [3], [7], [15]). Short-duration camps are particularly effective when designed to be interactive and relevant. University and community partnerships enhance sustainability and reach. Although successful models exist, they are disproportionately offered in high-resource areas. The literature reveals a gap in scalable outreach models for mid-sized or under resourced regions [2].

3.3 Instructional Strategies for Novice Learners

Instruction for novice learners must balance rigor with accessibility. Scaffolding and tool-based design reduce cognitive load and help students tackle complex concepts incrementally ([9], [16]). Tools such as Google Colab, Code.org, Scratch, and Autopsy support meaningful engagement with programming, artificial intelligence and machine learning, and cyberforensics without high technical barriers [3]. Gamification through progress tracking, challenge-based tasks, and low-stakes competition has been shown to increase persistence and engagement, especially in short-format learning environments ([4], [10], [19]).

AI-powered tools such as ChatGPT and GitHub Copilot are emerging instruction support tools that offer real-time content generation, scaffolding, and curriculum design, primarily in postsecondary and professional contexts [11]. Although its use in grades K-12 remains limited, initial studies suggest potential benefits for personalized support and instructional efficiency [13]. Students are engaged because they have been naturally curious about AI or using it with limited guidance, so this is early in the literature about using AI with K-12, by the students. However, educators must critically assess AI-generated content for accuracy and developmental appropriateness to avoid misinformation or confusion.

In sum, early, project-based computing experiences can be transformative for high school learners when grounded in real-world tools and accessible design. Informal models like summer camps can build early momentum toward computing pathways, though implementation in mid-sized regions and the use of AI tools in K–12 outreach remains underexplored. This case study addresses these gaps directly.

4 Design

To ensure accessibility, each camp was priced at $49, covering materials and refreshments. Funding from The 21st Century Partnership, a nonprofit supporting Robins Air Force Base and STEM development enabled the program. Outreach targeted motivated high school students, even those with no prior computing experience.

Each camp met for four hours daily over four consecutive days and served between 8 and 15 students. Registration was first-come, first-served, and intentionally limited to maintain a high instructor-to-student ratio. Students came from a mix of public, private, and homeschool backgrounds, and ranged in age from 13 to 18.

4.1 Python Programming Camp

Google Colab served as the primary platform to eliminate installation barriers and support collaborative, browser-based coding ([5], [18]). Instruction followed a cognitive framing approach, which refers to structuring content in a way that incrementally builds understanding by connecting new information to prior knowledge [12].

The four-day curriculum introduced variables, conditionals, loops, functions, and data structures including lists, sets, and dictionaries. Daily activities included debugging challenges ("Bug Hunts"), games ("Number Guessing Game"), and creative tools like a "Pizza Party Planner" and "Password Generator." Students also completed a playlist randomizer and used ChatGPT to build a calculator, illustrating how AI supports rapid prototyping. These tasks combined real-world logic, playful design, and incremental skill building.

The camp concluded with a competitive HackerRank challenge covering key topics. Students solved curated problems to demonstrate mastery, providing a gamified capstone aligned with core learning objectives [20].

4.2 Cyberforensics Camp

This camp introduced students to digital investigation through four themed days: forensic artifacts, investigative tools, timeline reconstruction, and reporting. Day 1 involved creating Windows 10 virtual machines, conducting traceable activities, then exchanging machines for peer analysis using Autopsy and FTK Imager. This gamified exercise emphasized evidence recovery and behavioral inference.

Day 2 added metadata analysis and steganography. Students uncovered hidden messages in files and dissected spoofed emails. The "Steganography Challenge" rewarded speed and creativity, while humorous narrative exercises, such as solving the case of a stolen Mona Lisa or a rogue foot-sword purchase, encouraged playful critical thinking.

Day 3 centered on OSINT, where students researched themselves, peers, and public figures. This sparked conversations around privacy, digital footprints, and ethics. They also learned email forensics and participated in a Wayground-hosted trivia game.

On Day 4, students analyzed EXIF metadata to trace geolocation, ultimately identifying a building in Serbia. The final challenge required building a forensic timeline and delivering a formal report. The camp concluded with a career talk covering certifications, job roles, and ethical responsibilities in cybersecurity.

Instructional design emphasized real-world tools, narrative framing, and flexible pacing. Mini-challenges, collaborative analysis, and ethics-driven scenarios supported high engagement, although many students needed individual support, highlighting both the potential and the limitations of the short-format model.

4.3 Artificial Intelligence/Machine Learning (AI/ML) Camp

The AI/ML camp guided students through four stages: foundational concepts, prompt engineering, app development, and model training. Activities emphasized ethical reflection, creative exploration, and technical skill-building.

On Day 1, students explored generative AI through story creation, vacation planning, and GUI design using tkinter and pygame. They used natural language prompts to build simple games, demonstrating how AI tools can assist novice programmers.

Day 2 focused on refining prompts. Students composed music in Code.org's Music Lab, completed an AI ethics module, and created interactive apps in App Lab. These tasks prompted discussions about authorship, bias, and responsible technology use.

On Day 3, students worked with digital media tools like Canva, Recraft.ai, OpenArt, Google Veo, and Suno AI, raising questions about attribution, authenticity, and synthetic content.

Day 4 shifted to model development using Code.org's AI and Machine Learning curriculum. Students trained classifiers on real-world datasets and used Teachable Machine to explore image, audio, and pose recognition. Final projects blended technical practice with critical reflection on AI's societal implications.

Although not explicitly organized around the AI4K12 framework, the camp addressed its core ideas. Students encountered *perception* through multimodal inputs, *representation and reasoning* via simple classifiers, *learning* through model training, and *natural interaction* through prompt engineering. Ethical discussions connected activities to AI's broader *societal impact.* The camp prioritized accessibility and inquiry, encouraging students to approach AI technologies with curiosity and critical awareness.

While each camp focused on distinct topics, Python programming, cyberforensics, or AI/ML, all shared a common instructional approach. They used scaffolded progression, gamified challenges, real-world tools, ethical inquiry, and career contextualization. Table 1 summarizes these strategies across camps.

Table 1: Instructional Design Elements Implemented Across All Camps

Instructional Element	**Implementation Across Camps**
Scaffolded Progression	Concepts and tools were introduced incrementally, with each day building on the last to support novice learners.
Gamification	Challenges such as trivia, HackerRank problems, steganography races, and creative competitions sustained engagement.
Real-World Tools	Students used authentic platforms like Google Colab, Autopsy, AI Lab, App Lab, ChatGPT, and Teachable Machine.
Ethics and Reflection	Each camp included discussions on privacy, authorship, and responsible technology use.
Career Awareness	Camps included exposure to job roles and ethical dilemmas, helping students view computing as creative and impactful.

5 Results

5.1 Student Participation and Engagement

Students were highly engaged across all camps, participating in hands-on activities that emphasized real-world tools and problem-solving. In the forensics camp, participants created virtual machines, conducted peer investigations, analyzed EXIF data, and compiled formal forensic reports. Python students tackled debugging challenges and developed interactive tools, while AI/ML students explored prompt engineering, generative code, and model development using AI Lab and Teachable Machine. Open-ended tasks such as OSINT research and digital crime scene analysis elicited especially strong interest and growing independence by Day 3.

A cognitive scaffolding approach proved effective for novice learners. Structured progression in Python supported incremental skill development, while AI-assisted tools enabled students with limited experience to produce functional applications. Although alternative pedagogical models exist, this approach offered a reliable foundation for learners with diverse backgrounds. The results suggest that, with accessible tools and thoughtful pacing, students in such regions can thrive in advanced computing environments.

Survey responses reflected this positive engagement. All participants reported increased knowledge of AI, cybersecurity, or programming. More than half indicated a significantly stronger interest in future study or careers in the field and expressed enthusiasm for participating in additional computing opportunities.

5.2 Instructional Effectiveness

Narrative framing and gamification were especially impactful. Activities like the Steganography Challenge, MS Paint contest, Wayground phishing game, HackerRank problems, and AI prompt contests reinvigorated groups and encouraged friendly competition. Students responded well to scenario-based tasks and peer collaboration, reinforcing both technical and real-world understanding. Assessment varied by camp but emphasized mastery: investigative reasoning in forensics, syntax and algorithmic thinking in Python, and prompt engineering and model accuracy in AI/ML.

5.3 Struggles and Variability

Despite strong engagement, students encountered challenges with multi-step instructions and unfamiliar tools. In forensics, these included software setup and operating system navigation. Python students faced issues with Colab and syntax, while AI/ML participants struggled with abstract concepts and lacked mathematical background. Some thrived independently; the majority required frequent instructor support. By Day 2, it was clear the pace was ambitious for several students.

Participants ranged from 8th to 12th grade, with wide variation in digital literacy, attention span, and learning affinity. Sustaining focus for four hours required redirection and occasional breaks. Some students had only used Apple devices and found it challenging to adjust to Windows, including saving files and navigating folder structures. A few believed they had “broken the computer" when executing terminal commands, highlighting how unfamiliar these environments were for some learners.

While progress was slower than anticipated, this diversity reinforced the need for structured, game-based instruction to support a broad range of experience levels.

5.4 Final Deliverables

Most students completed capstone projects. In forensics, they analyzed evidence, demonstrated EXIF analysis and OSINT skills and created professsional reports. In Python, students finished seven HackerRank problems aligned to camp content. In AI/ML, students presented culminating projects and reflected on AI's real-world impact. Task completion varied, particularly among less confident students or those discouraged by open-ended expectations.

6 Discussion

6.1 Engagement and Learning Gains

This pilot reinforces that short-format computing camps can spark interest in advanced topics among novice learners. Scenario-based inquiry, real-world tools, and gamified challenges effectively engaged students while introducing foundational computing skills. The approach aligns with prior research on informal STEM learning and supports project-based pre-college instruction.

Well-designed, high-rigor camps are both feasible and impactful in mid-sized regions like Middle Georgia. With interactive instruction and AI-supported tools, students without prior experience made meaningful progress. As computing becomes increasingly central to modern careers, regional access initiatives can help close opportunity gaps.

6.2 Challenges in Short-Format Camps

While camps offered valuable exposure, many students struggled with tool onboarding and task interpretation. Success hinged on structured materials, visuals, and peer support. For deeper learning, future efforts will require longer durations or repeated engagement over time.

6.3 Career Awareness and Perceptions

Many students began the camps with little to no familiarity with fields such as cybersecurity, artificial intelligence, or digital forensics. However, through exposure to investigative tools, ethical dilemmas, and hands-on, scenario-based projects, participants started to see these domains not simply as technical or abstract, but as creative, intellectually engaging, and socially relevant. This transformation highlights the critical role of intentional career framing in pre-college outreach efforts.

Several students expressed surprise at the range and nature of roles within cybersecurity, particularly noting that the work extended far beyond the common "hacker" stereotype. One participant remarked that the forensic activities felt "like being a digital detective," underscoring how experiential learning can challenge misconceptions and ignite interest in computing pathways.

Importantly, a lack of early, meaningful computing experiences often delays or discourages students from pursuing technical careers altogether. By demystifying complex topics and providing approachable entry points, these camps helped reframe students' understanding of the field and opened the door to continued exploration.

6.4 Instructional Design Takeaways

Pacing technical complexity and allowing time for exploration improved student outcomes. Low-penalty challenges, often described as "fail-soft" design, supported persistence by enabling learners to experiment and recover from mistakes without punitive consequences. This approach is particularly effective in short-format learning environments where confidence building is essential [4]. Because the camps relied primarily on no-cost tools and minimal setup requirements, they can be readily replicated at peer institutions. Activities were deliberately designed to be modular and interchangeable, using a wide variety of platforms to keep content fresh and adaptable. This modularity also provided resilience: it accounted for temporary student disengagement, the possibility of a tool malfunctioning, and the need to introduce diverse topics to sustain interest.

6.5 Limitations and Future Work

The four-day format raised awareness but limited content depth. Variation in student background presents replication challenges. Future programs could offer school-year extensions or modular follow-ups. Long-term evaluation will be essential to assess sustained gains in computational thinking and career orientation. Though post-program survey participation was limited, responses were overwhelmingly positive, signaling strong satisfaction and interest in future offerings.

7 Conclusion

This case study presents a transferable framework that bridges national AI/CS education initiatives with localized outreach models, offering both scholars and practitioners evidence that short-format camps can spark sustained computing engagement. By focusing on cyberforensics, artificial intelligence, and Python programming, the camps provided high school students in an under-resourced region with meaningful, hands-on experiences using real-world tools and novice-friendly instruction. Future iterations could extend this model by explicitly aligning with AI4K12 “big ideas,” complementing CSTA standards and AP CS Principles curricula.

Despite limited prior experience, students engaged enthusiastically with gamified challenges and structured support, closing awareness gaps and increasing interest in computing careers. As generative AI tools continue to evolve, these camps may further personalize instruction, foster creative exploration, and strengthen connections between technical skills and real-world applications.

While the four-day format constrained depth, the combination of scaffolded progression, AI-assisted instruction, and scenario-based learning fostered an inclusive and engaging environment. Sustained programming, curricular integration, or follow-up modules could deepen impact over time. Scaling initiatives through partnerships with school districts, nonprofit funders, and state-level CS education efforts offers a promising pathway to democratize access to advanced computing and reduce regional opportunity gaps.

Short-format computing camps, when designed with low-cost tools, gamified challenges, and scaffolded instruction, provide a replicable and scalable model for expanding access to advanced computing in under-resourced regions.

References

[1] Lecia J. Barker and William Aspray. *The state of research on girls and IT*. The MIT Press, Feb. 2006. ISBN: 9780262255929. DOI: https://doi.org/10.7551/mitpress/7272.003.0003.

[2] Code.org, CSTA, and the ECEP Alliance. *The state of computer science education: 2023 snapshot*. Annual Technical Report. Includes national and state-level data on K–12 CS access and equity. Code.org Advocacy Coalition, 2023. URL: https://advocacy.code.org/stateofcs/2023_state_of_cs.pdf.

[3] Jill Denner, Linda Werner, and Ed Ortiz. “The role of youth program settings in building STEM career interest”. In: *Afterschool Matters* 20 (2014), pp. 25–32. URL: https://files.eric.ed.gov/fulltext/EJ1046171.pdf.

[4] Christo Dichev and Darina Dicheva. “Gamifying education: What is known, what is believed and what remains uncertain: A critical review”. In: *International Journal of Educational Technology in Higher Education* 14.1 (2017), pp. 1–36. DOI: https://doi.org/10.1186/s41239-017-0042-5.

[5] Katlyn Edwards et al. “Google Colab for teaching CS and ML”. In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*. SIGCSE 2024. Portland, OR, USA: Association for Computing Machinery, 2024, p. 1925. ISBN: 9798400704246. DOI: https://doi.org/10.1145/3626253.3635432.

[6] Scott Freeman et al. “Active learning increases student performance in science, engineering, and mathematics”. In: *Proceedings of the National Academy of Sciences* 111.23 (2014), pp. 8410–8415. DOI: https://doi.org/{10.1073/pnas.1319030111}.

[7] Joanna Goode. “If you build teachers, will students come? The role of teachers in broadening computer science learning for urban youth”. In: *Journal of Educational Computing Research* 36 (Mar. 2007), pp. 65–88. DOI: https://doi.org/10.2190/2102-5G77-QL77-5506.

[8] Shuchi Grover and Roy Pea. “Computational thinking in K–12: A Review of the state of the field”. In: *Educational Researcher* 42.1 (2013), pp. 38–43. DOI: https://doi.org/10.3102/0013189X12463051.

[9] Shuchi Grover, Roy Pea, and Stephen Cooper. “Teaching CS Principles in an inquiry-based learning environment”. In: *ACM Transactions on Computing Education* 14.2 (2014), pp. 1–31. DOI: https://doi.org/10.1145/2602485.

[10] Juho Hamari, Jonna Koivisto, and Harri Sarsa. “Does gamification work? A literature review of empirical studies on gamification”. In: *Proceedings of the 47th Hawaii International Conference on System Sciences* (2014), pp. 3025–3034. DOI: https://doi.org/10.1109/HICSS.2014.377.

[11] Wayne Holmes, Maya Bialik, and Charles Fadel. *Artificial intelligence in education. Promise and implications for teaching and learning.* Center for Curriculum Redesign, Mar. 2019, p. 242. ISBN: 978-1794293700. URL: https://curriculumredesign.org/our-work/artificial-intelligence-in-education/.

[12] Yasmin B. Kafai and Chris Proctor. “A revaluation of computational thinking in K–12 education: Moving toward computational literacies”. In: *Educational Researcher* 51.2 (2022), pp. 146–151. DOI: https://doi.org/10.3102/0013189X211057904.

[13] Sang Joon Lee and Kyungbin Kwon. “A systematic review of AI education in K–12 classrooms from 2018 to 2023: Topics, strategies, and learning outcomes”. In: *Computers and Education: Artificial Intelligence* 6 (2024), p. 100211. DOI: https://doi.org/10.1016/j.caeai.2024.100211.

[14] National Research Council. *Learning science in informal environments: People, places, and pursuits*. Ed. by Philip Bell et al. Washington, DC: National Academies Press, 2009. DOI: https://doi.org/10.17226/12190.

[15] Florence R. Sullivan and Marina U. Bers. "Exploring computer science: A case study of school reform". In: *ACM Transactions on Computing Education* 15.2 (2015), pp. 1–31. DOI: https://doi.org/10.1145/2729986.

[16] John Sweller. "Cognitive load during problem solving: Effects on learning". In: *Cognitive Science* 12.2 (1988), pp. 257–285. DOI: https://doi.org/10.1207/s15516709cog1202_4.

[17] David Touretzky et al. "Envisioning AI for K-12: What should every child know about AI?" In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 9795–9799. DOI: https://doi.org/10.1609/aaai.v33i01.33019795.

[18] William Vallejo, Carlos Díaz-Uribe, and Catalina Fajardo. "Google Colab and virtual simulations: Practical e-learning tools to support the teaching of thermodynamics and to introduce coding to students". In: *ACS Omega* 7.8 (2022), pp. 7421–7429. DOI: https://doi.org/10.1021/acsomega.2c00362.

[19] Johnathan Yerby et al. "Development of serious games for teaching digital forensics". In: *Issues in Information Systems* 15.2 (2014). DOI: https://doi.org/10.48009/2_iis_2014_335-343.

[20] Kevin K. F. Yuen, Dennis Y. W. Liu, and Hong Va Leong. "Competitive programming in computational thinking and problem solving education". In: *Computer Applications in Engineering Education* 31.4 (2023), pp. 850–866. DOI: https://doi.org/10.1002/cae.22610.