

Python for All: A Library Workshop for Bridging AI Literacy and Coding Skills

Kristen L. Scotti and Lencia McKee*

In response to a growing, cross-disciplinary demand for individuals to develop programming skills, libraries are expanding their educational offerings to provide coding support for learners from diverse backgrounds. Teaching learners how to use generative artificial intelligence (AI) chatbots to code can enhance independent learning via real-time assistance and debugging support. Here, we describe the design, delivery, and assessment of a library workshop that teaches participants to leverage AI chatbots for learning Python. Python for All: Democratizing Coding Mastery with AI Chatbot Support combines AI literacy with practical coding skills to empower participants to use AI tools effectively and ethically. Workshop materials are available as open educational resources to support the democratization of coding education.

Introduction

The development and integration of generative artificial intelligence (GenAI) into education is reshaping the field (Lund et al., 2023; Walter, 2024). GenAI chatbots can personalize learning and offer real-time, iterative learning support (Ayala, 2023; Kasneci et al., 2023; Wu & Yu, 2023). However, GenAI integration also poses challenges; educators must manage increased risks of academic dishonesty (Hoq et al., 2024; Teel et al., 2023), address the potential for over-reliance on GenAI (Hoq et al., 2024), and tackle ethical concerns around data privacy and algorithmic bias (Ellis et al., 2024; Lund et al., 2023). These challenges are not entirely new. Since the 1950s, artificial intelligence (AI) has steadily integrated into education, evolving from computer-assisted instructional systems in the 1960s (Carbonell, 1980; Gable & Page, 1980), to intelligent tutoring systems in the 1970s (Anderson et al., 1995), and into dialogue-based tutoring systems (N. Kim et al., 1989), as well as exploratory learning environments (Hsu et al., 1993) in subsequent decades. Computers, the internet, and online learning brought similar challenges, requiring new pedagogical approaches and critical discussions about equitable access and privacy. Our need to balance innovation with ethics, access, and sustainability will continue as our educational practices and environments continue evolving.

The rapid integration of GenAI into educational settings makes AI literacy — that is, understanding, effectively using, and critically evaluating the impacts of AI technologies — imperative

*Kristen L. Scotti is a STEM Librarian at Carnegie Mellon University, email: kscotti@andrew.cmu.edu; Lencia McKee is Research Data Librarian at Cornell University, email: lcb235@cornell.edu. At the time this work was completed, McKee was affiliated with Carnegie Mellon University. ©2026 Kristen L. Scotti and Lencia McKee, Attribution-NonCommercial (<https://creativecommons.org/licenses/by-nc/4.0/>) CC BY-NC.

(Long & Magerko, 2020; Ng et al., 2021). AI literacy is intrinsically linked with information literacy (Bridges et al., 2024; James & Filgo, 2023), a foundational library service that teaches students how to identify, locate, evaluate, and use information effectively (Mackey & Jacobson, 2011). Libraries have consistently and repeatedly adapted to new technologies and consequent shifts in modes and practices of information sharing and have helped students to develop skills for engaging with information effectively and ethically (Walter, 2024). With the growing demand for programming skills and their increasing importance in understanding and leveraging AI technologies (Bridges et al., 2024), libraries have become increasingly responsible for providing coding education (Kang & Sinn, 2024; Martin, 2017). By integrating information literacy, AI literacy, and coding workshops, libraries can offer comprehensive educational programs that empower individuals to navigate this landscape more effectively, fostering a more informed and capable academic community.

This paper describes the design, implementation, and evaluation of Python for All: Democratizing Coding Mastery with AI Chatbot Support, a workshop offered by Carnegie Mellon University Libraries. This workshop blends hands-on exercises, guided learning, and group discussions to assist participants in developing both foundational programming skills and AI literacy. With this work, we aim to provide practical insights for educators and contribute to ongoing discussions about effective integration of AI tools in education.

Literature Review

Utilizing generative AI chatbots for teaching programming brings advantages and challenges. Artificial intelligence tools offer the potential to create more inclusive learning environments by providing instant feedback and better accommodating diverse educational needs and learning styles (relative to more traditional learning pedagogies) (Ayala, 2023). Multiple studies show that AI can enhance educational outcomes (Shen et al., 2024; Y.-C. Tsai, 2023; Zheng, 2023). Indeed, incorporating AI into programming instruction has been shown to increase student confidence and reduce anxiety (Becker et al., 2023; N. W. Kim et al., 2024; M.-L. Tsai et al., 2023; Y.-C. Tsai, 2023), a common impediment to learning how to code (Charles & Gwilliam, 2023; Demir, 2022; Özmen & Altun, 2014). However, studies have also shown negative impacts of AI integration. For example, an over-reliance on AI tools may impair students' problem-solving skill development (Ellis et al., 2024; Hoq et al., 2024; Joshi et al., 2024). Studies also highlight the limitations of AI tools. Haindl and Weinberger (2024) found that ChatGPT struggles with complex reasoning and code adjustments; their work emphasizes the need for human oversight and critical evaluation. Generative AI chatbots also have a tendency to confidently generate incorrect information (referred to as "hallucinations" or "fabrications") (Ahmad et al., 2023; Walters & Wilder, 2023). Good prompt engineering (the process of crafting effective prompts that generate desired results and mitigate AI inaccuracies and bias) is essential for maximizing the benefits of using GenAI for code generation (Denny et al., 2023; Ma et al., n.d.; Prather et al., 2024). Overall, studies underscore the importance of using AI as a complementary tool rather than a replacement for traditional learning methods.

Studies investigating the integration of GenAI chatbots into general programming education to teach programming skills are well-documented in the literature (Haindl & Weinberger, 2024; Hartley et al., 2024; Joshi et al., 2024; Ma et al., n.d.; T. Wang et al., 2024). However, we were unable to locate similar reports within the library and information science literature. Libraries recognize the importance of AI literacy, with literature in the field emphasizing the

need to teach students how to engage with AI tools responsibly (Cox & Tzoc, 2023; *IFLA Statement on Libraries and Artificial Intelligence*, n.d.; Lo, 2023a; Polverini & Gregorcic, 2024; Walter, 2024). Related case reports describe efforts by libraries to build AI literacy by teaching about AI tools and providing infrastructure to support their use (B. Kim, 2019; Michalak, 2024). Teaching with large language models (LLMs), such as ChatGPT (Johnson et al., 2024; Torres, 2024), the implementation of professional development programs for librarians to enhance their AI literacy skills (Lo, 2024), and using ChatGPT to develop educational resources (Cox & Tzoc, 2023) have also been documented. Despite these reports, empirical studies that investigate the impact of AI in library pedagogy are limited. In a study carried out in early 2024, Torres explored how GenAI models have been incorporated within library pedagogy and found that most of the published studies are conceptual. At the time of this writing, few empirical evaluations exist, which highlights a significant gap in understanding how GenAI tools, including chatbots, might be gainfully employed within library-facilitated coding workshops (wherein participants often represent more diverse programming backgrounds relative to students enrolled in computer science courses).

Workshop Description

The Python for All: Democratizing Coding Mastery with AI Chatbot Support workshop is aligned strategically with Carnegie Mellon University (CMU) libraries' commitment to fostering AI literacy (Bongiovanni et al., 2024; Slayton, 2025), and supporting open science initiatives (H. Wang et al., 2022). The libraries offer a diverse selection of workshops (CMU Libraries, 2024), including those that address fundamental programming skills, open science practices, and research data management. Within this landscape, the Python for All workshop targets individuals with some, but minimal coding knowledge, and aims to equip participants with the foundational skills needed to learn independently through GenAI chatbots, fostering self-reliance and adaptability in their programming journey. Thus, the focus on syntax is limited as this is addressed directly in other library workshops.

Over a two-hour virtual session, participants become familiar with AI concepts and vocabulary, use AI chatbots to generate code, and practice troubleshooting corresponding errors. To build participants' skills in assessing the accuracy and reliability of AI-generated code, we place a strong emphasis on critically evaluating the generated code. The workshop learning objectives are aimed at developing both technical and critical thinking skills:

- *AI Fundamentals*: Remember and understand key concepts and terminology related to GenAI by defining and describing them.
- *Practical Application*: Apply GenAI as a programming assistant to enhance coding efficiency and diagnose coding errors; demonstrate increased confidence in using GenAI tools by integrating GenAI to complete coding exercises.
- *Critical Evaluation*: Evaluate the accuracy, reliability, and usefulness of AI-generated code and solutions in coding projects.
- *Ethical and Practical Evaluation*: Analyze the ethical implications of generative AI, including its strengths, weaknesses, biases, and limitations.
- *Adaptability and Continuous Learning*: Create strategies to continuously adapt and respond to advancements in AI technologies, developing new skills to stay current.

Workshop materials are provided as open educational resources (OER; <https://osf.io/2xz7u/>), including introductory slides and student and instructor Jupyter notebooks. The

instructor notebook includes detailed notes for each exercise, covering learning objectives, teaching methods, and worked solutions. For teaching workshops, we keep the student Jupyter notebook in a GitHub repository and have participants access the notebook using MyBinder (Corbi et al., 2023). MyBinder is an open-source web service that allows users to execute notebooks (from GitHub) in a web browser; thus, participants are not required to install software on their personal devices prior to attending a workshop. Google Colab provides similar functionality (Bisong, 2019).

Exercises

Introduction to Prompt Engineering

Prompt engineering involves writing prompts in a manner that elicits desirable responses from AI models (Ekin, 2023). The *Python for All* workshop includes six exercises that collectively aim to build participants' prompt engineering skills; however, the first three exercises approach prompt engineering explicitly. We start with a description of the CLEAR framework (Concise, Logical, Explicit, Adaptive, and Reflective) for prompt engineering (Lo, 2023b), which provides a structured approach for improving interactions with GenAI models. In the first exercise, participants are asked to adopt the persona of a software engineer tasked with creating an algorithm to find the shortest path in a 2D maze. Using the CLEAR framework, they draft prompts with the objective of better understanding the logic behind the algorithm they are tasked with creating. As our participants typically come from diverse programming backgrounds, with many having little or no coding experience, we emphasize understanding programming logic over syntax to promote an understanding of foundational programming concepts that are necessary for critically evaluating generated code.

In the second exercise, participants use identical prompts to ask their chatbots to create a markdown table to help participants understand Python syntax elements. Upon comparing their results, they typically note differences among their peers' tables. We explain that these differences can result from training data variations, randomness in AI responses, and user-specific interactions (Bansal et al., 2024). By exploring the variability in AI responses, this exercise aims to enhance AI literacy, helping participants understand the complexities behind AI-generated responses.

In the final exercise of this section, participants generate code to produce a basic scatter plot in Python. We ask participants to brainstorm ideas for what to include in the prompts. Next, participants engage in an iterative process to modify various aspects of the plot (e.g., changing the marker colors). We encourage participants to review their code and try to identify which sections are relevant for making these changes. Participants are also guided through the process of using their chatbot to explain the code, line by line, and for generating code comments suitable for specified audiences. These steps ensure that participants grasp both the underlying logic and the syntax of the generated code, fostering both AI literacy and practical coding skills.

Simplifying Complex Coding Problems

Following the prompt engineering exercises, participants are asked to use their chatbots to build a number guessing game where the computer randomly selects a number for a user to guess. This exercise builds on the algorithmic logic introduced in the first exercise (where participants are asked to conceptualize a problem prior to generating code). Here, participants

deconstruct the multifaceted coding challenge (i.e., building a number guessing game) into manageable parts using pseudocode. Pseudocode uses plain language to outline the steps of an algorithm, again allowing participants to focus on logic over syntax (Bellamy, 1994), while also providing a basis for writing subsequent prompts (T. Wang et al., 2024). We begin with a complex problem statement and demonstrate the first step in breaking down the problem into parts, writing the first identified task (i.e., generating a random number) in pseudocode. Participants are subsequently guided through creating a prompt that will produce code for the task. After demonstrating the first step, participants are grouped into breakout rooms to work together to identify remaining tasks, craft corresponding prompts, and test resulting snippets individually; ultimately, they combine their snippets to test functionality of the final product. Breaking down complex tasks into parts reinforces algorithmic literacy while also helping to improve the accuracy of generated code (Haindl & Weinberger, 2024). Moreover, working with code snippets simplifies debugging processes, making it easier to identify and correct issues along the way (Sadowski et al., 2018).

Code Optimization and Translation

The last two exercises teach participants to enhance code efficiency and adapt it to different programming languages. In the first exercise, participants work with a set of provided Python functions that sum all prime numbers below an input number with the goal of reducing the computational resources needed for the task. Many algorithms exist for this problem, each differing in effectiveness based on a given scenario (e.g., input value; Crandall & Pomerance, 2001). The diversity of potential solutions facilitates conversations about potential AI biases when optimizing code—participants learn about the importance of algorithmic optimization and explore how AI might favor certain algorithms or libraries based on its training data (Ferrara, 2023).

Participants begin by running the provided code to verify its functionality while ensuring that they fully understand the original solution prior to proceeding. Next, they are tasked with asking their chosen chatbot to optimize the code for better performance. The key part of the learning process involves critically evaluating whether the chatbot suggested optimization actually improves efficiency (and if so, under which conditions it does so) (Liu et al., 2024). To accomplish this, participants engage in hands-on testing, where they compare execution times of both the original and optimized versions of the code, using various input values.

The last exercise involves translating MATLAB (a proprietary language) code to Python (an open-source language). We provide MATLAB code that generates a plot; the code includes a “hold” attribute that is valid in MATLAB but deprecated in Python’s Matplotlib library. Chatbots sometimes retain this attribute during translation; if they do, running the code will return an error. These errors provide an opportunity to discuss limitations that arise from outdated training data (Torres, 2024). Chatbots also tend to correct the plot title from “Plot of the Sine Function” (as specified in the original MATLAB code) to “Plot of the Sine and Cosine Functions” (based on the context of the original code, which plots both sine and cosine functions). While this adjustment increases the accuracy of the resulting figure, it raises important questions about AI systems making unprompted changes. This leads to a discussion on whether such automatic corrections are appropriate and whether they align with the user’s original intent. Participants are encouraged to critically assess AI-generated code to ensure it accurately reflects their specific goals and maintains both clarity and correctness.

Workshop Summary

The pedagogical techniques employed in the Python for All workshop are designed to teach information and AI literacy competencies through hands-on, iterative learning and critical evaluation of AI-generated code outputs. The exercises employ best practices for effective learning, including immediate feedback, practical application of theoretical concepts, and collaborative problem-solving (Chi & Wylie, 2014; Hattie & Timperley, 2007; Rummel & Spada, 2005). For example, the prompt engineering exercises encourage students to refine their prompts, understand AI variability, and enhance AI literacy, reflecting findings from previous studies (Polverini & Gregorcic, 2024; Walter, 2024) that emphasize the importance of understanding LLM mechanisms and prompt engineering. The exercise that requires simplifying coding problems aims to strengthen participants' algorithmic literacy skills, addressing the need for human oversight in complex reasoning tasks (Haindl & Weinberger, 2024). Finally, the code optimization and translation exercises address potential limitations in using AI to generate code, encouraging critical evaluation of AI output (Backström & Kihlert, 2023; Rahman & Watanobe, 2023). Together, these exercises aim to provide participants with an understanding of GenAI models, while fostering critical thinking, adaptability, and proficiency in leveraging these models to enhance their programming skills.

Evaluation

We used pre- and post-surveys via Zoom to evaluate the effectiveness of the workshop in enhancing participants' confidence in using GenAI chatbots to learn Python. The pre-survey, given at the beginning of each workshop, contained three multiple choice questions that assessed participants' previous experience with Python, GenAI chatbots, and using GenAI chatbots for coding tasks, specifically, as well as free response questions for providing their role within the university and their academic field. Five options are provided for each question, ranging from no experience to expert-level experience for each of these skills. The post-survey, administered at the end of each workshop, included two multiple choice questions assessing participants' likelihood of using generative AI chatbots for coding in the future (with five options, ranging from very unlikely to very likely) and rating any confidence changes due to the workshop (again, five options, ranging from "significantly decreased confidence" to "significantly increased confidence"). The post-survey also contained a free-response question, inviting participants to provide feedback on the workshop. The surveys were intentionally kept short to facilitate administration during the workshop; thus, we did not assess each learning objective individually. Instead, the pre-survey responses helped tailor the depth of code explanations that we provided during each workshop, and the post-survey was used to gauge the workshop's overall effectiveness in enhancing participant's confidence in using GenAI chatbots to continue learning Python.

This assessment was reviewed and determined to be exempt by the Institutional Review Board at Carnegie Mellon University under 2018 Common Rule 45 CFR 46.104(d). All workshop participants were informed that their responses to survey questions were voluntary and anonymous. No identifying personal information was requested or collected.

Participants

The Python for All workshop attracted a diverse group of participants, possibly reflecting a widespread interest in using GenAI chatbots for learning code. A total of 93 participants

TABLE 1
Workshop Participant and Survey Respondent Backgrounds by Role and Academic Field
(Across Five Workshops)

	Participants	Completed Surveys
	n=93	n=62
Role		
Undergraduate Students	1 (1%)	1 (2%)
Graduate Students	58 (62%)	37 (60%)
Postdoctoral Researchers	2 (2%)	2 (3%)
Staff	14 (15%)	10 (16%)
Faculty	14 (15%)	10 (16%)
Community Members	4 (4%)	2 (3%)
Academic Field		
Biology, Chemistry	6 (7%)	5 (8%)
Business	17 (18%)	10 (16%)
Computer Science	14 (15%)	10 (16%)
Engineering	10 (11%)	9 (15%)
Fine Arts	5 (5%)	2 (3%)
Public Policy	39 (42%)	25 (40%)
Psychology	2 (2%)	1 (2%)

attended one of five two-hour workshop sessions (offered monthly between June and October 2024), including (62%) graduate students, 28 (30%) faculty and staff members, four (4%) community members, two (2%) postdoctoral researchers, and one (1%) undergraduate student. Participants represented various academic fields, with 42% representing public policy, 18% from business, 15% from computer science, and others from biology, chemistry, engineering, fine arts, and psychology (see Table 1). The relatively high representation of public policy participants can be attributed to targeted outreach by the liaison librarian to Carnegie Mellon's Heinz College of Information Systems and Public Policy. Faculty were also well represented, at 15% of participants, which is higher than usual for similar workshops. Increased faculty involvement likely reflects a growing interest in exploring methods for AI chatbot integration in curricula. Some faculty participants reached out for further support, aligning with broader trends previously reported (Miller, 2024), where instructors are reaching out to their librarians for guidance on managing and incorporating GenAI tools in their teaching.

Outcomes

Of 93 total workshop participants, 62 (67%) fully completed both the pre- and post-workshop survey. As previously stated, all participants were informed that responding to survey questions was both voluntary and anonymous. The pre-survey results showed a range of Python experience levels. Specifically, 26% of participants had no prior Python experience, 34% rated their experience at a beginner level, 31% as intermediate, and 6% and 3% as advanced and expert-level, respectively. More notably, responses regarding previous use of GenAI chatbots

emphasize the growing importance of integrating AI literacy into education; only 11% of participants reported no prior experience with generative AI chatbots (i.e., the overwhelming majority [89%] already had some level of experience with these tools). About 10% of participants reported little previous experience, 31% reported occasional use, and 48% reported regular or frequent usage. Additionally, 13% of participants reported frequent use of generative AI chatbots for coding tasks, specifically. As we grapple to find the balance between discouraging use of GenAI and promoting responsible engagement (Lau & Guo, 2023; Miller, 2024), it is important to recognize that students are likely already engaging with these technologies. This lends credibility to the argument that we have a responsibility to equip students with the knowledge necessary to engage ethically with these tools (Borenstein & Howard, 2021), regardless of whether educators or institutions ultimately seek to restrict or encourage their use.

The heatmap in Figure 1 shows the relationship between participants' (n = 62) self-rated pre-workshop experience in Python and their post-workshop confidence change for using generative AI chatbots to enhance their coding skills. Most participants (58%) reported that their post-workshop confidence "significantly increased;" about 35% reported their post-workshop confidence as "increased," and 6% reported no change. No participants reported a decrease in post-workshop confidence. Participants with no prior Python experience reported the highest post-workshop confidence increases, with 69% reporting a significant increase in confidence, while those with advanced and expert-level Python experience reported lesser gains (see Figure 1, Table 2). As our workshop is targeted toward beginner-level Python participants, these results provide some assurance that we are meeting the needs of that group. Importantly, as our post-workshop survey was conducted immediately after workshop sessions, we are unable to assess whether these gains translated to longer term confidence gains.

FIGURE 1
Heat Map Illustrating the Relationship Between Participants' (n = 24) Pre-Workshop Experience in Python (top) and Changes in Post-Workshop Confidence for Using Generative AI Chatbots to Enhance Python Coding Skills.

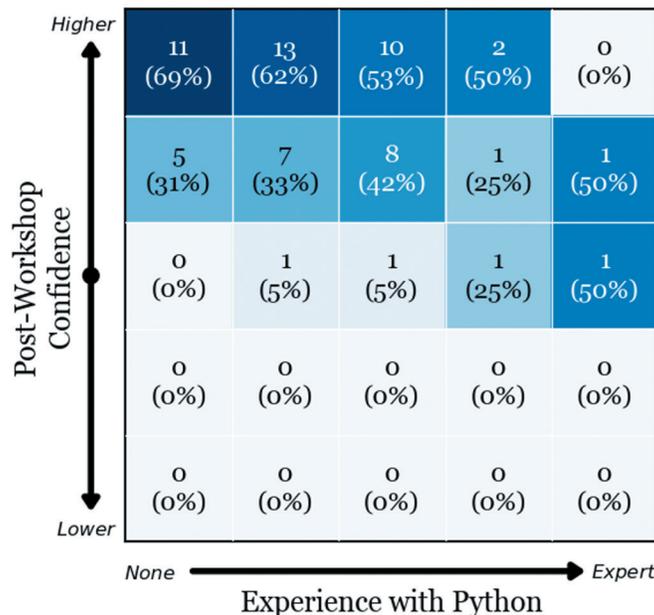


TABLE 2
Changes in Reported Post-Workshop Confidence Levels Using GenAI Chatbots to Learn Python

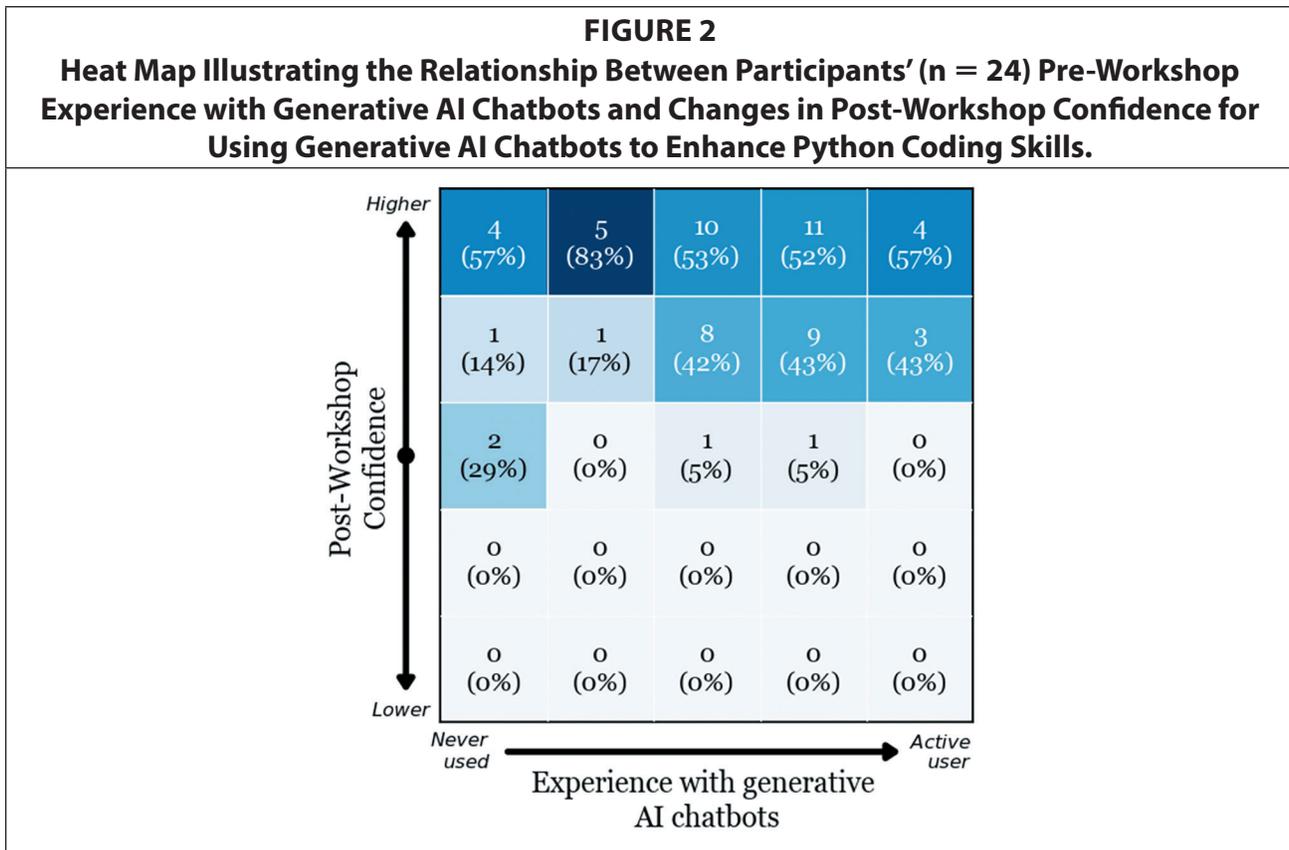
	Participants	Significantly Decreased Confidence	Decreased Confidence	No Change	Increased Confidence	Significantly Increased Confidence
Python Skill Level						
No Experience	16 (26%)	0	0	0	5 (31%)	11 (69%)
Beginner	21 (34%)	0	0	1 (5%)	7 (33%)	13 (62%)
Intermediate	19 (31%)	0	0	1 (5%)	8 (42%)	10 (53%)
Advanced	4 (6%)	0	0	1 (25%)	1 (25%)	2 (50%)
Expert	2 (3%)	0	0	1 (50%)	1 (50%)	0
Chatbot Experience						
No Experience	7 (11%)	0	0	2 (29%)	1 (14%)	4 (57%)
Tried it Once or Twice	6 (10%)	0	0	0	1 (17%)	5 (83%)
Occasional User	19 (31%)	0	0	1 (5%)	8 (42%)	10 (53%)
Regular User	22 (35%)	0	0	1 (5%)	9 (43%)	12 (52%)
Frequent User	8 (13%)	0	0	0	3 (43%)	5 (57%)

Figure 2 provides a heatmap showing the relationships between participants' self-rated pre-workshop experience using generative AI chatbots (for coding or otherwise) and their post-workshop confidence changes in leveraging AI chatbots for coding applications in the future. Participants with some, but minimal, prior experience using generative AI chatbots reported the highest post-workshop confidence gains with 83% reporting a significant increase in confidence. However, participants' previous experience with chatbots is not highly correlated overall with post-workshop confidence.

Participant feedback provided within the free response portion of the survey was almost exclusively positive; participants expressed appreciation for the workshop's hands-on approach, and four participants requested more exercises as "homework." Two respondents indicated that the pacing of the workshop was too fast. Overall, these outcomes suggest that the workshop effectively enhanced participants' technical skills and confidence in using AI-assisted coding tools.

Challenges and Future Directions

Initial iterations of this workshop (not included in the evaluative metrics provided here) focused on using GenAI chatbots to translate MATLAB code to Python. Our motivation was to promote the use of open-source tools (e.g., Python) as part of a broader effort to support open science initiatives. Low registration numbers suggested that our scope was too narrow. We expanded the workshop to more broadly cover the use of GenAI chatbots for coding in Python, which increased interest. Shifting to a virtual format offered improved engagement as the online platform made it easier to facilitate a collaborative learning environment as participants were better able to share their screens.



During the first workshop, participants found the number guessing game exercise too challenging. In subsequent workshop sessions, we included more emphasis on basic programming skills within the preliminary exercises. We also added breakout rooms to encourage peer-learning. Participant feedback indicated that breakout room format helped them work through the problem-solving process. During breakout sessions, participants asked questions, shared their screens to compare outputs, and experimented with different approaches for obtaining AI-generated code. To encourage collaboration and reflection, each group was asked to nominate a “spokesperson” to report back to the larger group on how they approached the problem and what they learned. One group noted that their AI model generated code that provided the user hints if they guessed a number that was lower or higher than the computer-generated number, even though the prompt did not provide those instructions. Other groups discussed adjusting prompts to improve outputs or to prevent the model from anticipating next steps. Another group explored whether prompts worked the same with numbers typed out (e.g., “ten”) versus numerals (e.g., “10”). Across the four (of five) workshops for which these approaches were implemented, every breakout room group completed all tasks.

Looking ahead, we plan to refine the workshop content to address varying levels of programming and AI experience, including developing preparatory materials for beginners to review prior to the workshop. We are also considering expanding the workshop into a series to cover more advanced topics. Continuous assessment and participant feedback will remain integral for iteratively improving the workshop to meet evolving needs.

Conclusion

The Python for All: Democratizing Coding Mastery with AI Chatbot Support workshop teaches AI literacy using Python programming instruction. The aim of the workshop is to teach participants how to use AI tools in learning how to code in Python. The workshop integrates hands-on exercises, peer-learning, and opportunities for critical discussions that collectively promote an understanding of applications and limitations of AI-generated code. We used pre- and post-workshop surveys to evaluate the effectiveness of the workshop; survey results over five workshop sessions indicate that the workshop increased participants' confidence for using generative AI chatbots to learn coding skills. Participants with minimal previous coding experience reported the highest post-workshop confidence gains. By documenting the workshop's structure, learning objectives, and outcomes, this paper contributes to the broader conversation on AI integration in education. The workshop materials are available as an OER to support educators with integrating AI literacy into curricula.

Acknowledgments

The authors thank Melanie Gainey, PhD, for her contributions to the conceptualization of the workshop and for providing thoughtful feedback on the draft of this paper. We also thank Sarah Young, MLS, for her dedicated outreach efforts in participant recruitment.

References

- Ahmad, Z., Kaiser, W., & Rahim, S. (2023). Hallucinations in ChatGPT: An unreliable tool for learning. *Rupkatha Journal on Interdisciplinary Studies in Humanities*, 15. <https://doi.org/10.21659/rupkatha.v15n4.17>
- Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, Ray. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 4(2), 167–207. https://doi.org/10.1207/s15327809jls0402_2
- Ayala, S. (2023). ChatGPT as a universal design for learning tool. *Educational Renaissance*, 12(1), 23–41.
- Backström, O., & Kihlert, A. (2023). Code quality and large language models in computer science education: Enhancing student-written code through ChatGPT.
- Bansal, G., Chamola, V., Hussain, A., Guizani, M., & Niyato, D. (2024). Transforming conversations with AI—A comprehensive study of ChatGPT. *Cognitive Computation*, 16(5), 2487–2510. <https://doi.org/10.1007/s12559-023-10236-2>
- Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., & Santos, E. A. (2023). *Programming is hard-or at least it used to be: Educational opportunities and challenges of AI code generation*. 500–506.
- Bellamy, R. (1994). What does pseudo-code do? A psychological analysis of the use of pseudo-code by experienced programmers. *Human-Computer Interaction*, 9(2), 225–246. https://doi.org/10.1207/s15327051hci0902_3
- Bisong, E. (2019). Google colab. In E. Bisong, *Building machine learning and deep learning models on Google cloud platform* (pp. 59–64). Apress. https://doi.org/10.1007/978-1-4842-4470-8_7
- Bongiovanni, E., Slayton, E., Agate, N., Flierl, M., Lan, H., Scotti, K., Tatarian, A., Young, A., Janco, A., & Ferer, E. (2024). AI literacy resource hackathon. *Open Science Framework*. <https://doi.org/10.17605/OSF.IO/WS2CE>
- Borenstein, J., & Howard, A. (2021). Emerging challenges in AI and the need for AI ethics education. *AI and Ethics*, 1(1), 61–65. <https://doi.org/10.1007/s43681-020-00002-7>
- Bridges, LM., McElroy, K., & Welhouse, Z. (2024). Generative artificial intelligence: 8 critical questions for libraries. *Journal of Library Administration*, 64. <https://doi.org/10.1080/01930826.2024.2292484>
- Carbonell, J. R. (1980). AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, 11(4), 190–202.
- Charles, T., & Gwilliam, C. (2023). The effect of automated error message feedback on undergraduate physics students learning Python: Reducing anxiety and building confidence. *Journal for STEM Education Research*, 6(2), 326–357. <https://doi.org/10.1007/s41979-022-00084-4>
- Chi, M. T. H., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist*, 49(4), 219–243. <https://doi.org/10.1080/00461520.2014.965823>
- CMU Libraries. (2024). *Workshops | CMU Libraries*. <https://www.library.cmu.edu/services/workshops>
- Corbi, A., Burgos, D., & Pérez, A. M. (2023). Cloud-operated open literate educational resources: The case of the MyBinder. *IEEE Transactions on Learning Technologies*. <https://ieeexplore-ieee-org.cmu.idm.oclc.org/abstract/document/10365663/>

- Cox, C., & Tzoc, E. (2023). ChatGPT: Implications for academic libraries. *College and Research Libraries News*, 84. <https://doi.org/10.5860/crln.84.3.99>
- Crandall, R., & Pomerance, C. (2001). *Prime numbers*. Springer. <https://doi.org/10.1007/978-1-4684-9316-0>
- Demir, F. (2022). The effect of different usage of the educational programming language in programming education on the programming anxiety and achievement. *Education and Information Technologies*, 27(3), 4171–4194. <https://doi.org/10.1007/s10639-021-10750-6>
- Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B. A., & Reeves, B. N. (2023). Promptly: Using prompt problems to teach learners how to effectively utilize ai code generators. *arXiv Preprint arXiv:2307.16364*.
- Ekin, S. (2023). Prompt engineering for ChatGPT: A quick guide to techniques, tips, and best practices. *Authorea Preprints*.
- Ellis, M. E., Casey, K. M., & Hill, G. (2024). ChatGPT and Python programming homework. *Decision Sciences Journal of Innovative Education*, 22(2), 74–87.
- Ferrara, E. (2023). Fairness and bias in artificial intelligence: A brief survey of sources, impacts, and mitigation strategies. *Sci*, 6(1), 3.
- Gable, A., & Page, C. V. (1980). The use of artificial intelligence techniques in computer-assisted instruction: An overview. *International Journal of Man-Machine Studies*, 12(3), 259–282.
- Haindl, P., & Weinberger, G. (2024). Students' experiences of using ChatGPT in an undergraduate programming course. *IEEE Access*, 12, 43519–43529.
- Hartley, K., Hayak, M., & Ko, U. H. (2024). Artificial intelligence supporting independent student learning: An evaluative case study of ChatGPT and learning to code. *Education Sciences*, 14(2), 120.
- Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, 77(1), 81–112. <https://doi.org/10.3102/003465430298487>
- Hoq, M., Shi, Y., Leinonen, J., Babalola, D., Lynch, C., Price, T., & Akram, B. (2024). Detecting ChatGPT-generated code submissions in a CS1 course using machine learning models. 526–532.
- Hsu, J.-F. J., Chapelle, C. A., & Thompson, A. D. (1993). Exploratory learning environments: What are they and do students explore? *Journal of Educational Computing Research*, 9(1), 1–15. <https://doi.org/10.2190/VLPO-EC65-GBT5-32D4>
- IFLA statement on libraries and artificial intelligence. (n.d.).
- James, A. B., & Filgo, E. H. (2023). Where does ChatGPT fit into the Framework for Information Literacy? The possibilities and problems of AI in library instruction. *College and Research Libraries News*, 84(9), 334–341. Scopus. <https://doi.org/10.5860/crln.84.9.334>
- Johnson, S., Owens, E., Menendez, H., & Kim, D. (2024). Using ChatGPT-generated essays in library instruction. *Journal of Academic Librarianship*, 50(2). <https://doi.org/10.1016/j.acalib.2024.102863>
- Joshi, I., Budhiraja, R., Dev, H., Kadia, J., Ataulloh, M. O., Mitra, S., Akolekar, H. D., & Kumar, D. (2024). ChatGPT in the classroom: An analysis of its strengths and weaknesses for solving undergraduate computer science questions. 625–631.
- Kang, G., & Sinn, D. (2024). Technology education in academic libraries: An analysis of library workshops. *Journal of Academic Librarianship*, 50(2), N.PAG-N.PAG. Scopus. <https://doi.org/10.1016/j.acalib.2024.102856>
- Kasneji, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., ... Kasneji, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103. <https://doi.org/10.1016/j.lindif.2023.102274>
- Kim, B. (2019). AI and creating the first multidisciplinary AI lab. *Library Technology Reports*, 55(1), 16.
- Kim, N., Evens, M., Michael, J. A., & Rovick, A. A. (1989). Circsim-tutor: An intelligent tutoring system for circulatory physiology. In H. Maurer (Ed.), *Computer assisted learning* (Vol. 360, pp. 254–266). Springer. https://doi.org/10.1007/3-540-51142-3_64
- Kim, N. W., Ko, H.-K., Myers, G., & Bach, B. (2024). ChatGPT in data visualization education: A student perspective. *arXiv Preprint arXiv:2405.00748*.
- Lau, S., & Guo, P. (2023). From “ban it till we understand it” to “resistance is futile”: How university programming instructors plan to adapt as more students use AI code generation and explanation tools such as ChatGPT and GitHub Copilot. 106–121.
- Liu, J., Xia, C. S., Wang, Y., & Zhang, L. (2024). Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36.
- Lo, L. (2023a). An initial interpretation of the U.S. Department of Education's AI report: Implications and recommendations for academic libraries. *Journal of Academic Librarianship*, 49(5), N.PAG-N.PAG. Ixh. <https://doi.org/10.1016/j.acalib.2023.102761>

- Lo, L. (2023b). The CLEAR path: A framework for enhancing information literacy through prompt engineering. *Journal of Academic Librarianship*, 49(4). <https://doi.org/10.1016/j.acalib.2023.102720>
- Lo, L.S. (2024). Transforming academic librarianship through AI reskilling: Insights from the GPT-4 exploration program. *Journal of Academic Librarianship*, 50. <https://doi.org/10.1016/j.acalib.2024.102883>
- Long, D., & Magerko, B. (2020). What is AI literacy? Competencies and design considerations. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. <https://doi.org/10.1145/3313831.3376727>
- Lund, B.D., Wang, T., Mannuru, N.R., Nie, B., Shimray, S., & Wang, Z. (2023). ChatGPT and a new academic reality: Artificial intelligence-written research papers and the ethics of the large language models in scholarly publishing. *Journal of the Association for Information Science and Technology*, 74. <https://doi.org/10.1002/asi.24750>
- Ma, B., Li, C., & Shin'ichi, K. (n.d.). *Enhancing programming education with ChatGPT: A case study on student perceptions and interactions in a Python course*. Retrieved May 23, 2024, from <https://arxiv.org/html/2403.15472v3>
- Mackey, T. P., & Jacobson, T. E. (2011). Reframing information literacy as a metaliteracy. *College & Research Libraries*, 72(1), 62–78.
- Martin, C. (2017). Libraries as facilitators of coding for all. *Knowledge Quest*, 45(3), 46–53.
- Michalak, R. (2024). Fostering undergraduate academic research: Rolling out a tech stack with AI-powered tools in a library. *Journal of Library Administration*, 64(3), 335–346. Scopus. <https://doi.org/10.1080/01930826.2024.2316523>
- Miller, R. E. (2024). Pandora's can of worms: A year of generative AI in higher education. *Portal: Libraries and the Academy*, 24(1), 21–34. <https://doi.org/10.1353/pla.2024.a916988>
- Ng, D. T. K., Leung, J. K. L., Chu, S. K. W., & Qiao, M. S. (2021). Conceptualizing AI literacy: An exploratory review. *Computers and Education: Artificial Intelligence*, 2, 100041. <https://doi.org/10.1016/j.caeai.2021.100041>
- Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming: Difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3). <https://doi.org/10.17569/tojqi.20328>
- Polverini, G., & Gregorcic, B. (2024). How understanding large language models can inform the use of ChatGPT in physics education. *European Journal of Physics*, 45(2), 025701.
- Prather, J., Denny, P., Leinonen, J., Smith IV, D. H., Reeves, B. N., MacNeil, S., Becker, B. A., Luxton-Reilly, A., Amarouche, T., & Kimmel, B. (2024). Interactions with prompt problems: A new way to teach programming with large language models. *arXiv Preprint arXiv:2401.10759*.
- Rahman, M. M., & Watanobe, Y. (2023). ChatGPT for education and research: Opportunities, threats, and strategies. *Applied Sciences*, 13(9), 5783.
- Rummel, N., & Spada, H. (2005). Learning to collaborate: An instructional approach to promoting collaborative problem solving in computer-mediated settings. *Journal of the Learning Sciences*, 14(2), 201–241. https://doi.org/10.1207/s15327809jls1402_2
- Sadowski, C., Söderberg, E., Church, L., Sipko, M., & Bacchelli, A. (2018). Modern code review: A case study at Google. *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, 181–190. <https://doi.org/10.1145/3183519.3183525>
- Shen, Y., Ai, X., Soosai Raj, A. G., Leo John, R. J., & Syamkumar, M. (2024). *Implications of ChatGPT for data science education*. 1230–1236.
- Slayton, E. (2025). Developing a primer for developing data skills: The story behind Carnegie Mellon University Libraries data literacy program. In *Data culture in academic libraries: A practical guide to building communities, partnerships, and collaborations*.
- Teel, Z. (Abbie), Wang, T., & Lund, B. (2023). ChatGPT conundrums: Probing plagiarism and parroting problems in higher education practices. *College and Research Libraries News*, 84(6). <https://doi.org/10.5860/crln.84.6.205>
- Torres, J. (2024). Leveraging ChatGPT and bard for academic librarians and information professionals: A case study of developing pedagogical strategies using generative AI models. *Journal of Business and Finance Librarianship*. Scopus. <https://doi.org/10.1080/08963568.2024.2321729>
- Tsai, M.-L., Ong, C. W., & Chen, C.-L. (2023). Exploring the use of large language models (LLMs) in chemical engineering education: Building core course problem models with Chat-GPT. *Education for Chemical Engineers*, 44, 71–95.
- Tsai, Y.-C. (2023). *Empowering learner-centered instruction: Integrating ChatGPT Python API and tinker learning for enhanced creativity and problem-solving skills*. 531–541.
- Walter, Y. (2024). Embracing the future of artificial intelligence in the classroom: The relevance of AI literacy, prompt engineering, and critical thinking in modern education. *International Journal of Educational Technology in Higher Education*, 21(1), 15. <https://doi.org/10.1186/s41239-024-00448-3>
- Walters, W.H., & Wilder, E.I. (2023). Fabrication and errors in the bibliographic citations generated by ChatGPT. *Scientific Reports*, 13. <https://doi.org/10.1038/s41598-023-41032-5>

- Wang, H., Gainey, M., Campbell, P., Young, S., & Behrman, K. (2022). Implementation and assessment of an end-to-end open science and data collaborations program. *F1000 Research*, 11, 501. Scopus. <https://doi.org/10.12688/f1000research.110355.2>
- Wang, T., Zhou, N., & Chen, Z. (2024). Enhancing computer programming education with LLMs: A study on effective prompt engineering for Python code generation. *arXiv Preprint arXiv:2407.05437*. [https://arxiv-org.cmu.idm.oclc.org/abs/2407.05437](https://arxiv.org/cmu.idm.oclc.org/abs/2407.05437)
- Wu, R., & Yu, Z. (2023). Do AI chatbots improve students learning outcomes? Evidence from a meta-analysis. *British Journal of Educational Technology*, 55. <https://doi.org/10.1111/bjet.13334>
- Zheng, Y. (2023). *ChatGPT for teaching and learning: An experience from data science education*. 66–72.