Published: 05 September 2024

UKICER 2024: The United Kingdom and Ireland Computing Education Research
September 5 - 6, 2024
Manchester, United Kingdom

RESEARCH-ARTICLE

Exploring Human-Centered Approaches in Generative AI and Introductory Programming Research: A Scoping Review

IRENE STONE, Dublin City University, Dublin, Leinster, Ireland

UKICER '24: Proceedings of the 2024 Conference on United Kingdom & Ireland Computing Education Research (September 2024)
https://doi.org/10.1145/3689535.3689553
ISBN: 9798400711770

Exploring Human-Centered Approaches in Generative AI and Introductory Programming Research: A Scoping Review

Irene Stone
Dublin City University
Ireland
irene.stone2@mail.dcu.ie

Abstract

Recent advancements in generative artificial intelligence are poised to reshape introductory programming education, challenging conventional teaching methodologies. This paper presents a scoping review that explores the current understanding of integrating generative artificial intelligence tools in the learning of introductory programming. Through an analysis of 28 selected studies, this review provides a snapshot of the landscape in mid-2024, presenting benefits, concerns, and recommendations surrounding the use of generative artificial intelligence within programming education. It finds insufficient guidance on how to implement recommended pedagogical strategies, limited consideration of student perceptions and experiences, and a predominance of short study time frames. Additionally, there is a significant research gap in second-level education, particularly in the United Kingdom and Ireland. The paper discusses how these gaps signal a need for more human-centered approaches in the current research. The paper concludes with recommendations for future research, aiming to inspire further inquiry and advance the understanding of generative artificial intelligence's role in programming education from a human-centered perspective.

CCS Concepts

• Computing methodologies → Artificial intelligence; **• Social and professional topics → Computing education**; **CS1**; **Computer science education**.

Keywords

AI; artificial intelligence; ChatGPT; code generation; CS1; generative AI; human-centered; learner perspectives; LLMs; novice programming; pedagogical practices; programming; python; student-centered

ACM Reference Format:
Irene Stone. 2024. Exploring Human-Centered Approaches in Generative AI and Introductory Programming Research: A Scoping Review. In *The United Kingdom and Ireland Computing Education Research (UKICER 2024), September 05–06, 2024, Manchester, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3689535.3689553

UKICER 2024, September 05–06, 2024, Manchester, United Kingdom

ACM ISBN 979-8-4007-1177-0/24/09
https://doi.org/10.1145/3689535.3689553

1 Introduction

The landscape of programming education shifted dramatically with the release of generative artificial intelligence (GenAI) tools like ChatGPT and Copilot in 2022. These tools, which can generate code, are highly capable of solving problems typical of introductory programming courses and often outperform students in code writing [11]. With Computer Science students actively using these tools in their studies [32], this prompts a closer examination of challenges and opportunities in introductory programming education.

As part of a doctoral research journey, a scoping review was conducted on 3rd May 2024. The aim was to identify gaps in knowledge [26] within the field of GenAI and introductory programming. Employing a framework proposed by Arksey and O'Malley [3], this scoping review draws from various databases to gain comprehensive insights. As a secondary school teacher and in line with UNESCO's invitation for educators and researchers to prioritise human agency when using GenAI [24], the review adopts a human-centered approach. The paper advocates for a paradigm that ensures the active participation and voices of human participants.

The primary contributions of this paper include:

- Providing an overview of benefits, concerns, and recommendations for GenAI use in learning programming.
- Identifying gaps that exist in the current body of literature.
- Proposing recommendations for integrating human-centred approaches into research on GenAI tools in introductory programming education.

2 Methodology

Scoping reviews are ideal for identifying knowledge gaps, scoping a body of literature, and clarifying concepts [26]. This review employs a well-established and widely-utilised five-stage framework for scoping reviews [3]:

- Stage 1: Identifying the research questions
- Stage 2: Identifying relevant studies
- Stage 3: Study selection
- Stage 4: Charting the data
- Stage 5: Collating, summarising and reporting the results

2.1 Stage 1: Identifying the research questions

The following research questions guide the search and the review:

RQ1 What is known from existing literature regarding GenAI and the learning of introductory programming?
RQ2 What gaps exist in the current body of literature?

Table 1: Search Criteria

"introductory programming" AND "generative artificial intelligence"
(cs1 OR "foundations of programming" OR "fundamental programming" OR "introductory programming" OR "novice programming") AND ("chat gpt" OR "chatgpt" OR "chatgpt-3" OR "chatgpt-3.5" OR "chatgpt-4" OR "copilot" OR "openai" OR "generative artificial intelligence" OR "genai" OR "large language models" OR "llm")

2.2 Stage 2: Identifying relevant studies

A systematic search was conducted across the ACM Digital Library, EBSCOhost, and Web of Science (WoS) databases. The search aimed to identify relevant peer-reviewed literature on GenAI in introductory programming education. The primary topics of interest are "introductory programming," and "generative artificial intelligence". In identifying relevant studies, these two terms serve as the foundation for the search criteria. Other relevant keywords or terminologies are extracted from these terms, as detailed in Table 1.

Given the novelty of the GenAI topic, the search was restricted to content published within the last 2 years. Concentrating solely on peer-reviewed articles, the search strings were applied to all fields. A total of 179 results were retrieved, including 30 duplicates. Four articles were irretrievable and three articles were excluded due to being less than two pages long. Subsequently, 142 articles were identified for screening at the title and abstract level.

2.3 Stage 3: Study Selection

The screening process involved reviewing all 142 abstracts. This led to the identification of 28 empirical research articles for the review, all of which involve human participants. The exclusion of 114 articles was due to various reasons.

- Assessing the capabilities of GenAI tools: 38 papers did not involve human participants.
- Papers not focusing on GenAI or introductory programming: 36 papers appeared in the search results due to mention of the relevant search criteria terms in their "related work" or "future work" sections.
- Irrelevant contexts: 14 papers primarily focus on the development and/or evaluation of AI-powered technology tools in areas such as creative-coding environments, virtual avatars, e-books, automated assessment tools, and web-based games.
- Non-empirical papers: 15 papers were excluded as they refer to opinion pieces and lack relevant empirical data.
- Various other reasons: 11 papers were excluded due to reasons including plagiarism (6 papers), technical focus (3 papers), industry relevance (1 paper), and focus on teaching about AI (1 paper).

The process of article selection followed the well-established Preferred Reporting of Items for Systematic Reviews and Meta-Analyses (PRISMA) format [29] as illustrated in Fig. 1.

2.4 Stage 4: Charting the data

A “data charting form” was created using an Excel spreadsheet. The 28 papers are categorised and presented in Table 2.

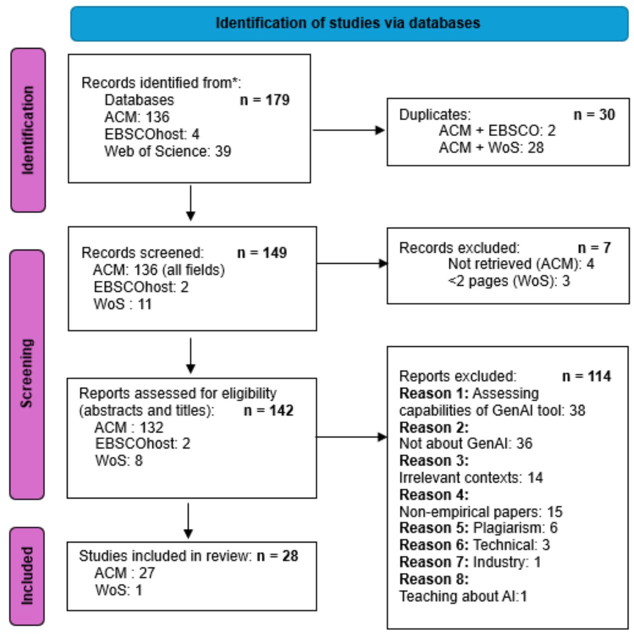

Figure 1: PRISMA chart for study selection

2.5 Stage 5: Collating, summarising and reporting the results

The final step in the five-stage framework involves collating, summarising, and reporting the scoping review’s findings. This is presented in the next section.

3 Results

This section addresses the two research questions through a comprehensive review of the 28 selected papers. Q1 is addressed by analysing the papers to identify benefits, concerns, and recommendations related to GenAI and the learning of introductory programming. Q2 is explored by identifying gaps in the literature.

3.1 RQ1: What is known from the existing literature?

Benefits: The literature reveals a multitude of positive aspects associated with the integration of GenAI tools in programming education: As previously mentioned, papers assessing the capabilities of GenAI tools were excluded from the review due to the lack of human involvement in the research (Fig. 1). However, other

Table 2: Details of selected papers (n = number of participants)
***Duration not applicable in studies primarily involving surveys and/or interviews.**
**** These two studies refer to the same controlled experiment. *** Both papers are based on the same research study.**

Author(s)	Type of study	Duration*	Data Collection	Country	Level	Proficiency	n
[1]	Human experiment	40mins	Static Code Analy., Eye-tracking	USA	3rd	Mixed	21
[2]	Exploratory study	N/A*	Survey	USA, India	3rd	Mixed	253
[5]	Grounded Theory	1 hour	Observation, Interview	unknown	Mixed	Mixed	20
[6]	Mixed Methods Research Design	N/A	Survey Interview	India	3rd	Mixed	480 (17 interview)
[8]	Controlled experiment	3 weeks	Questionnaires, Log data	USA	3rd	Mixed	22
[9]	Development of ART tool	N/A	Survey	USA	3rd	Intermediate	74
[10]	Development of "Promptly" tool	Single class session	Prompt problem responses, open-ended survey responses	New Zealand	3rd	Mixed	240 (153 survey)
[14]	Exploratory	N/A	Survey, Interview	Nth America	3rd	Mixed	47(8 interview)
[15]	Investigative workshop study	2hour	Video recordings, interview, code, reflections	Sweden	3rd	Introductory	21
[16]**	Controlled Experiment	3 weeks	Pre & Post tests, Survey	Canada	2nd	Introductory	69
[17]**	Controlled Experiment	3 weeks	Log data	Canada	2nd	Introductory	69
[19]	Development "KOGI" tool	4 weeks	Interactions, weekly feedback	Japan	3rd	Intermediate	127
[20]	Empirical study gathering instructor perspectives	N/A	Interview	Int'l	3rd (Instructors)	Advanced	20
[21]	Evaluating code explanations (Human vs AI)	2 lab sessions, 2 wks apart	Comparing code explanations & open-ended responses	NZ	3rd	Introductory	954 (100 survey)
[22]***	Evaluation of guardrails "CodeHelp" tool	12 weeks	Survey & usage analysis	USA	3rd	Introductory	52 (45 survey)
[23]	Development of "CS50.ai" tool	Summer & Fall '23	Usage logs & survey	USA	3rd	Introductory	570
[30]	Empirical study	N/A	Surveys & evaluation of ChatGPT code	UK	3rd Industry	Mixed	37 survey 17 assessment
[31]	Experience report	2 months	Prompt data, surveys	USA	3rd	Mixed	64 (52 survey)
[32]	Working group report	N/A	Lit review, surveys, interviews, datasets for benchmarking	Int'l	3rd students, instructors	Mixed	171 st., 57inst. (22 interview)
[33]	Observational Study	30 mins	Observations, Interviews	USA	3rd	Introductory	19
[34]	Experimental study	4 hours	Pre & post tests	Saudi Arabia	3rd	Intermediate	24
[35]	Qualitative study	2hr session	Focus groups	Canada	3rd students, instructors	Mixed	40
[36]	Development of "StAp-tutor" tool	1 hour	Student interactions, feedback, evaluation of tool, interview	Neth'lands	3rd	Introductory	3 students (+ 2 experts)
[38]	Empirical study	<1hour	Evaluation of screening questions, survey				121
[39]	Phenomenological approach	N/A	Interview	Australia, Finland, NZ	3rd (Instructors)	Advanced	12
[40]***	Evaluation of guardrails "CodeHelp" tool	12 weeks	Analysis of query logs	USA	3rd	Introductory	49
[41]	Experimental study	1hr per session	Test scores	UAE	3rd	Introductory	41 to 56
[42]	Evaluating performance & instructor perspectives	N/A	Performance data, survey,interview	USA	3rd (Instructors)	Mixed	11 instructors

studies involving human participants assessing the capabilities of GenAI tools were included in the review. Of these, notable benefits relating to GenAI tools emerge. AI-generated explanations surpass student-crafted ones in accuracy and understandability with novices preferring AI-generated detailed code explanations [21]. GenAI has the ability to match human-written code in complexity and readability [1], enhancing proficiency in various question types, including code analysis and conceptual questions [41].

GenAI can accelerate programming progress, enabling learners to work at a higher level of abstraction [33]. In "acceleration mode", programmers, who know their next steps, use Copilot to expedite the process, expanding their understanding of unfamiliar concepts, and in "exploration mode", it can help students who are uncertain about what to do next [5]. Similarly, ChatGPT can help students brainstorm [6] and initiate [35] ideas. Additionally, there is recognition of the time-saving potential of AI-generated code

[16, 30]. Students frequently use GenAI tools to seek programming help [2] and for debugging [6, 14, 32, 40]. Furthermore, GenAI tools are valuable for providing feedback [6, 14, 34]. GenAI can act as a co-creator by providing starter code and prompting users to expand and refine their ideas [15].

Various studies have introduced innovative tools integrated with GenAI to aid in the learning of programming. An Automatic Review Tool (ART) receives positive feedback for defect detection and style-based suggestions [9]. PROMPTLY, a web-based tool that hosts a repository of prompt problems, is designed to develop students' ability to prompt effectively and has been linked to the enhancement of Computational Thinking skills [10]. KOGI, a learning support system that seamlessly integrates ChatGPT into the Jupyter coding environment, showcases promising outcomes in reducing unresolved student errors [19]. A study involving 10 to 17-year-olds reveals the potential benefits of the AI-generated code tool "Coding Steps" for novice programmers, indicating its usefulness in reducing frustration and stress [16].

Concerns: While GenAI tools offer numerous benefits, concerns regarding their implementation in educational settings are voiced in the literature. These concerns revolve around the potential for incorrect or biased responses [8, 33–35, 41] as well as the risk of being led astray by inaccurate suggestions, a concept known as "drifting" [33].

A common theme that emerges from the literature is over-reliance [1, 5, 6, 10, 22, 33]. This could negatively impact job readiness [35]. Students struggle with writing prompts [10, 31, 40], yet often use a single prompt to generate entire solutions [17].

A recurring finding in the literature is the negative impact of GenAI on novice programmers [8, 14, 16]. Instructors have expressed concerns about introducing AI to students before they achieve proficiency in programming [39] or while they are still learning the basics [32]. Premature exposure to AI may lead to "shallow learning" among students [42]. While reducing frustration is seen as a benefit of using GenAI among novice programmers [17], other literature has shown increased frustration levels among those with mixed or intermediate proficiency levels [8, 34].

Academic integrity concerns surface [35, 41], particularly regarding the potential for plagiarism [39] and AI-assisted cheating [20]. Equity and access emerge as concerns for CS instructors, as the benefits and drawbacks of GenAI tools are not equally distributed among all students [20]. Other concerns include: reduced communication with instructors [35], logging of data [31] as well as concerns regarding reliability, accuracy, unethical use, and lack of humanistic qualities [6, 22].

Recommendations: To mitigate potential challenges associated with the use of GenAI tools in educational settings, several recommendations emerge from the literature. These include clear guidelines for ChatGPT use, a requirement for students to report usage, balancing in-class and take-home assessments, and fact-checking ChatGPT outputs [35]. Additionally, a cautious approach with warning messages is advised to remind students of GenAI fallibility, along with monitoring usage and interactions [22, 34].

Regarding novice programmers, instructors recommend disallowing the use of GenAI tools in lower-level courses [32], which highlights the necessity for careful consideration and selective implementation within this demographic [8, 14, 39]. Limits should be imposed on novice programmers when using GenAI so they are not led down incorrect solution paths [33].

In relation to pedagogical approaches, crafting effective prompts is key in using GenAI [6, 10, 15, 34]. Using keywords such as "student" or "hint" in prompts is suggested [36]. Developing help-seeking skills [14, 40] and decomposition skills is also essential and emphasising the importance of verifying AI-generated code is crucial [17]. Students should be educated on effective questioning techniques [41]. Other recommendations include introducing code review as a pedagogical approach [9] and teachers role-playing good and bad AI tool usage [22].

In terms of assessment, there is a recognised need for innovative methods that promote critical thinking [41]. Assessments could be developed to encourage learners to critically evaluate AI output [30]. Alternative assessments proposed include oral, and video, as well as process-based assessments [20, 39]. The concept of "process over product" is also suggested elsewhere, indicating a shift away from focusing solely on final solutions [32]. Interviews could be utilised to detect plagiarism, and hands-on learning should be encouraged [34]. Regarding exams, there is a growing trend favoring invigilated assessments [32]. To prevent users from easily generating meaningful prompts, avoiding extended answer options in multiple-choice questions is suggested and using visual questions to challenge AI [38]. This aligns with other research advocating for image-based assessments [20, 41]. However, with the release of ChatGPT-4o in May 2024, which can interpret images, further research is needed to develop robust strategies against AI-assisted cheating.

Concerning the AI models themselves, it is recommended they generate shorter code suggestions and readability should be incorporated into the model [1]. GenAI tools could be designed with pedagogical guardrails that don't disclose answers directly, rather, students are guided toward solutions rather than them being offered outright [22, 23] and code could be divided into multiple segments [16]. Future AI tools should recognise whether a programmer is in acceleration or exploration mode and offer suggestions with intentional gaps or "holes" to encourage programmers to actively engage with the generated code, promoting a deeper understanding and preventing over-reliance on the tool [5]. A summary of the benefits, concerns, and recommendations is presented in Table 3.

3.2 RQ2: What gaps exist in the literature?

Despite numerous recommendations regarding GenAI and programming (Table 3), significant gaps in the literature emerge:

Need for a Comprehensive Pedagogical Framework: There is a notable absence of a comprehensive pedagogical framework to use with GenAI in the teaching and learning of introductory programming. Focusing on *some* of the pedagogical recommendations as presented in Table 3, it is unclear how these could be implemented. For instance, *how* can educators help students develop skills in prompt writing, as advocated in the literature [6, 10, 15, 34]? Similarly, *how* can they help students develop decomposition skills [17], cultivate help-seeking skills [14, 40] or teach effective questioning techniques [41]? Just as PRIMM was developed to address the need for structured programming teaching approaches, particularly in school settings [37], a comprehensive framework is required to

Table 3: Benefits, Concerns, and Recommendations regarding GenAI and Introductory Programming
***Recommendations relating to making changes in AI models are not included**

Benefits of GenAI	Concerns around GenAI	Recommendations for educators*
Accelerate programming progress [5, 33]	Academic integrity, AI assisted cheating & plagiarism [20, 35, 39, 41]	Balance in-class and take-home assignments [35]
AI-generated code solutions match/ surpass human ones [1, 21]	Effects on novice programmers [8, 14, 16, 32, 39, 42]	Clear guidelines [35]
Brainstorming/ initiating ideas [6, 35]	Equity and access [20]	Develop decomposition[17], help-seeking[14, 40], prompt-writing skills[6, 10, 15, 34]
Debugging code [6, 14, 32, 40]	Frustration [8, 34]	Educate students on effective questioning techniques [41]
Promote creativity [15]	Hallucination (incorrect responses) or biased responses [8, 33–35, 41]	Emphasise process over product [20, 32, 39]
Providing starter code & scaffolding for novices [15, 16]	Impact negatively on job readiness [35]	Encourage hands-on learning [34]
Receiving feedback [6, 14, 34]	Leading students astray [33]	Fact-check/ verify outputs [17, 35]
Reducing frustration & stress [16]	Logging of data [31]	Impose limits on its use with novices [8, 14, 32, 39]
Reducing unresolved student errors [19]	Not humanistic [6, 22]	Innovative assessments: AI-proof exams, video/image-based, interviews[20, 34, 38, 41]
Seek programming help [2]	Over-reliance [1, 5, 6, 10, 22, 33]	Introduce code review as a practice [9]
Time saving [16, 30]	Reduced communication with instructors [35]	Monitor usage [22, 34]
	Struggle with writing prompts [10, 17, 31, 40]	Require students to report their AI usage [35]
		Teachers role-playing good and bad AI tool usage [22]
		Use of words "hint" and "student" in prompts [36]

support the integration of GenAI [25].

Focus on learners' perceptions and experiences: Although a significant proportion of the *selected* studies used methods conducive to exploring student perceptions and insights, such as surveys, interviews, focus groups, log data, reflections, [2, 6, 8–10, 14–16, 21–23, 30–33, 35, 36, 38], this represents a relatively small fraction of the 142 abstracts screened (Figure 1). This limited consideration of student-centered perspectives within the context of GenAI and introductory programming research is observed in the literature. Research on AI code-generation tools often lacks a focus on learning perspectives, particularly in terms of students' thought processes and motivations [16, 17] and their perceptions of these tools [2]. This scarcity of student-centered research [6] highlights neglect of "what students actually do" [31]. Much of the research tends to concentrate on AI tool performance [42]. While it is encouraging to see a gradual shift in the literature from solely assessing the capabilities of large language models [10], there remains a clear call for research to illuminate the intricacies of student experiences, perceptions, and interactions in this domain.

Introductory programming learners as research participants: While "introductory programming" was a search term in the initial stages of the review, Table 1, less than half of the selected studies solely involved introductory programmers, Table 2.

Need for long-term impact studies: Most of the studies were conducted over short time frames, limiting their ability to capture the long-term impacts on student learning and engagement. This limitation is also noted in the literature [31, 42].

Inclusion of diverse global perspectives: The prevalence of studies based in North America underscores the necessity for research from diverse global perspectives, particularly in the UK and Ireland.

Classroom-based studies in second-level education: There is a notable gap in understanding the effects of GenAI at other educational levels, particularly at second level—a domain experiencing international expansion in Computer Science education [18]. Notably, the single study from the scoping review addressing this age group was not conducted within a classroom setting [16]. With evidence of students struggling with writing prompts at the graduate level [10], it is crucial to investigate what this means in second-level education.

The next section discusses how these gaps signal a need for human-centered approaches to research in the area of introductory programming and GenAI, and makes recommendations for the research community on how to move in this direction.

4 Discussion

In exploring human-centered approaches in GenAI and introductory programming research, it is important to come to an understanding of what is meant by human-centered approaches in research. This concept is challenging to define, as it means "many different things to different people" [7]. Given that the focus of this paper is on learning, insights are drawn from UNESCO's *Guidance for Generative AI in Education and Research*, which emphasises that a human-centered approach requires researchers and educators to "prioritise human agency" when deciding on whether and how to use GenAI [24]. In the context of AI, human agency refers to users' ability to make informed, autonomous decisions regarding AI systems [12] and where AI should not be allowed to "usurp human thinking" [24]. In protecting human agency, UNESCO calls for consulting learners about their views on GenAI and recommends that its use be "co-designed by teachers, learners, and researchers" [24].

Adopting a human-centered approach means ensuring that students are not merely subjects of the research but active contributors. Their agency involves "acting rather than being acted upon; shaping rather than being shaped; and making responsible decisions and choices rather than accepting those determined by others" [27]. This point is supported by the European Commission, which emphasises that for the ethical use of AI in education, people should not be treated merely as a "data object" [13]. In practical terms, this necessitates actively engaging students by placing them at the core of the research process and dedicating sufficient time to facilitate their meaningful involvement. This involves conducting research within their real-world educational contexts and monitoring their interactions to better understand their experiences, challenges, and perspectives of GenAI.

The gaps highlighted in Section 3.2 underscore a critical need for human-centered approaches in GenAI and introductory programming research. The insufficient focus on learners' perspectives and experiences, and limited involvement of introductory programming learners highlight the need for a shift towards human-centered research. Incorporating learners' insights over longer timeframes, and involving participants in the process will help in the development of effective pedagogical strategies or frameworks. This will ensure that GenAI tools are integrated in a manner that supports students' autonomy and decision-making capabilities. In particular, classroom-based studies at the secondary level are needed to observe and understand the real-world interactions of students with GenAI, ensuring that research in GenAI and programming education is truly centered on the needs and experiences of learners.

Consequently, to adopt a human-centred approach in research on the use of GenAI in introductory programming, this paper makes the following recommendations for the research community:

- Actively involve students in the research process by seeking their input and feedback throughout.
- Conduct studies over longer-term periods to facilitate deeper engagement and allow for monitoring of their actions.
- Utilise participatory research methodologies

5 Limitations

This scoping review was conducted by a single author as part of their doctoral studies. Consequently, there may be reliability issues relating to the scoping review results and the consistency of exclusions. These concerns were mitigated by adhering to the Scoping Framework [3] and methodological PRISMA [29]. Furthermore, the author made efforts to substantiate the claims by referencing existing literature. Finally, given the rapid explosion of literature in this field, there is a risk that the findings may quickly become outdated.

6 Future Work

This paper underscores a critical gap in our understanding of how students engage with GenAI in introductory programming education. Without this understanding, the development of a pedagogical framework for integrating GenAI would be premature. Therefore, further research focused on exploring student interaction with GenAI is essential, especially among second-level students where human-centered research remains limited. While many countries, including Ireland [28], await guidelines for teaching and learning with GenAI, keeping student agency at the forefront when integrating GenAI is crucial.

The author proposes employing a design-based research (DBR) methodology with students in an Irish second-level school. This approach aligns with human-centered principles, by first focusing on students' perspectives and experiences using ChatGPT when learning Python. Next, students act as "co-participants" [4] in the iterative design of a framework to support Python learning with ChatGPT. The research will be situated in their context, the classroom.

To the author's knowledge, the proposed research study marks the first exploration of ChatGPT use by novice programmers among second-level students in a school setting.

7 Conclusion

The scoping review presented in this paper explored the evolving landscape of introductory programming education and the impact of GenAI tools. By analysing 28 studies, this paper provides a snapshot of the benefits, concerns, and recommendations for GenAI use in learning programming. It identifies gaps in the literature and discusses how human-centered approaches could be integrated into research in this area. To fill some of the existing research gaps, a design-based research methodology is proposed, to be conducted over time in a second-level setting. Feedback and connections with others interested in advancing the understanding of GenAI in educational programming contexts are welcome. The aim is to motivate discussion around human-centered approaches in this area.

Acknowledgments

I extend special thanks to Dr Margaret Leahy, DCU for her invaluable support and advice. Thanks also to the UKICER reviewers for their feedback.

References

[1] Naser Al Madi. 2023. How Readable is Model-generated Code? Examining Readability and Visual Inspection of GitHub Copilot. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering (ASE '22)*. ACM, New York, NY, USA, 1–5. https://doi.org/10.1145/3551349.3560438

[2] Matin Amoozadeh, David Daniels, Daye Nam, Aayush Kumar, Stella Chen, Michael Hilton, Sruti Srinivasa Ragavan, and Mohammad Amin Alipour. 2024. Trust in Generative AI among Students: An exploratory study. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024)*. ACM, New York, NY, USA, 67–73. https://doi.org/10.1145/3626252.3630842

[3] Hilary Arksey and Lisa O'Malley. 2005. Scoping studies: towards a methodological framework. *International Journal of Social Research Methodology* 8, 1 (Feb. 2005), 19–32. https://doi.org/10.1080/1364557032000119616

[4] Sasha Barab and Kurt Squire. 2004. Design-Based Research: Putting a Stake in the Ground. *Journal of the Learning Sciences* 13, 1 (Jan. 2004), 1–14. https://doi.org/10.1207/s15327809jls1301_1

[5] Shraddha Barke, Michael B. James, and Nadia Polikarpova. 2023. Grounded Copilot: How Programmers Interact with Code-Generating Models. *Proceedings of the ACM on Programming Languages* 7, OOPSLA1 (April 2023), 85–111. https://doi.org/10.1145/3586030

[6] Ritvik Budhiraja, Ishika Joshi, Jagat Sesh Challa, Harshal D. Akolekar, and Dhruv Kumar. 2024. "It's not like Jarvis, but it's pretty close!" - Examining ChatGPT's Usage among Undergraduate Students in Computer Science. In *Proceedings of the 26th Australasian Computing Education Conference* (Sydney, NSW, Australia,) *(ACE '24)*. ACM, New York, NY, USA, 124–133. https://doi.org/10.1145/3636243.3636257

[7] Tara Capel and Margot Brereton. 2023. What is Human-Centered about Human-Centered AI? A Map of the Research Landscape. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany,) *(CHI '23)*. ACM, New York, NY, USA, 1–23. https://doi.org/10.1145/3544548.3580959

[8] Rudrajit Choudhuri, Dylan Liu, Igor Steinmacher, Marco Gerosa, and Anita Sarma. 2024. How Far Are We? The Triumphs and Trials of Generative AI in Learning Software Engineering. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. ACM, New York, NY, USA, 1–13. https://doi.org/10.1145/3597503.3639201

[9] Aaron S. Crandall, Gina Sprint, and Bryan Fischer. 2023. Generative Pre-Trained Transformer (GPT) Models as a Code Review Feedback Tool in Computer Science Programs. *Journal of Computing Sciences in Colleges* 39, 1 (Oct. 2023), 38–47.

[10] Paul Denny, Juho Leinonen, James Prather, Andrew Luxton-Reilly, Thezyrie Amarouche, Brett A. Becker, and Brent N. Reeves. 2024. Prompt Problems: A New Programming Exercise for the Generative AI Era. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. ACM, New York, NY, USA, 296–302. https://doi.org/10.1145/3626252.3630909

[11] Paul Denny, James Prather, Brett A. Becker, James Finnie-Ansley, Arto Hellas, Juho Leinonen, Andrew Luxton-Reilly, Brent N. Reeves, Eddie Antonio Santos, and Sami Sarsa. 2024. Computing Education in the Era of Generative AI. *Commun. ACM* 67, 2 (Jan. 2024), 56–67. https://doi.org/10.1145/3624720

[12] European Commission. 2019. *Ethics guidelines for trustworthy AI.* Publications Office of the European Union, Luxembourg. https://op.europa.eu/en/publication-detail/-/publication/d3988569-0434-11ea-8c1f-01aa75ed71a1 OCLC: 1128948729.

[13] European Commission. 2022. *Ethical guidelines on the use of artificial intelligence (AI) and data in teaching and learning for educators.* Publications Office of the European Union, Luxembourg. https://data.europa.eu/doi/10.2766/153756

[14] Irene Hou, Sophia Mettille, Owen Man, Zhuo Li, Cynthia Zastudil, and Stephen MacNeil. 2024. The Effects of Generative AI on Computing Students' Help-Seeking Preferences. In *Proceedings of the 26th Australasian Computing Education Conference* (Sydney, NSW, Australia) *(ACE '24)*. ACM, New York, NY, USA, 39–48. https://doi.org/10.1145/3636243.3636248

[15] Martin Jonsson and Jakob Tholander. 2022. Cracking the code: Co-coding with AI in creative programming education. In *Proceedings of the 14th Conference on Creativity and Cognition* (Venice, Italy) *(C&C '22)*. ACM, New York, NY, USA, 5–14. https://doi.org/10.1145/3527927.3532801

[16] Majeed Kazemitabaar, Justin Chow, Carl Ka To Ma, Barbara J. Ericson, David Weintrop, and Tovi Grossman. 2023. Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. ACM, NY, USA, 1–23. https://doi.org/10.1145/3544548.3580919

[17] Majeed Kazemitabaar, Xinying Hou, Austin Henley, Barbara Jane Ericson, David Weintrop, and Tovi Grossman. 2023. How Novices Use LLM-based Code Generators to Solve CS1 Coding Tasks in a Self-Paced Learning Environment. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '23)*. ACM, New York, NY, USA, 1–12. https://doi.org/10.1145/3631802.3631806

[18] Neil Keane and Clare McInerney. 2017. *Report on the Provision of Courses in Computer Science in Upper Second Level Education Internationally*. Technical Report. National Council for Curriculum and Assessment (NCCA). https://ncca.ie/en/resources/computer_science_report_sc/

[19] Kimio Kuramitsu, Yui Obara, Miyu Sato, and Momoka Obara. 2023. KOGI: A Seamless Integration of ChatGPT into Jupyter Environments for Programming Education. In *Proceedings of the 2023 ACM SIGPLAN International Symposium on SPLASH-E* (Cascais, Portugal) *(SPLASH-E 2023)*. ACM, NY, USA, 50–59. https://doi.org/10.1145/3622780.3623648

[20] Sam Lau and Philip Guo. 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. In *Proceedings of the 2023 ACM Conference on International Computing Education Research V.1* (Chicago, IL, USA) *(ICER '23)*. ACM, New York, NY, USA, 106–121. https://doi.org/10.1145/3568813.3600138

[21] Juho Leinonen, Paul Denny, Stephen MacNeil, Sami Sarsa, Seth Bernstein, Joanne Kim, Andrew Tran, and Arto Hellas. 2023. Comparing Code Explanations Created by Students and Large Language Models. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1* (Turku, Finland) *(ITiCSE 2023)*. ACM, NY, USA, 124–130. https://doi.org/10.1145/3587102.3588785

[22] Mark Liffiton, Brad E Sheese, Jaromir Savelka, and Paul Denny. 2023. CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '23)*. ACM, New York, NY, USA, 1–11. https://doi.org/10.1145/3631802.3631830

[23] Rongxin Liu, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J. Malan. 2024. Teaching CS50 with AI: Leveraging Generative Artificial Intelligence in Computer Science Education. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. ACM, New York, NY, USA, 750–756. https://doi.org/10.1145/3626252.3630938

[24] Fengchun Miao and Wayne Holmes. 2023. *Guidance for generative AI in education and research | UNESCO*. UNESCO. https://www.unesco.org/en/articles/guidance-generative-ai-education-and-research

[25] Punya Mishra, Melissa Warr, and Rezwana Islam. 2023. TPACK in the age of ChatGPT and Generative AI. *Journal of Digital Learning in Teacher Education* 39, 4 (Oct. 2023), 235–251. https://doi.org/10.1080/21532974.2023.2247480

[26] Zachary Munn, Micah D. J. Peters, Cindy Stern, Catalin Tufanaru, Alexa McArthur, and Edoardo Aromataris. 2018. Systematic review or scoping review? Guidance for authors when choosing between a systematic or scoping review approach. *BMC Medical Research Methodology* 18, 1 (Dec. 2018), 143. https://doi.org/10.1186/s12874-018-0611-x

[27] OECD. 2019. *OECD Learning Compass 2030: A series of Concept Notes*. Technical Report. Organisation for Economic Co-operation and Development. https://www.oecd.org/content/dam/oecd/en/about/projects/edu/education-2040/1-1-learning-compass/OECD_Learning_Compass_2030_Concept_Note_Series.pdf

[28] Department of Education. 2024. Minister Foley pledges commitment to establishing guidelines on the use of AI. https://www.gov.ie/en/press-release/87b43-minister-foley-pledges-commitment-to-establishing-guidelines-on-the-use-of-ai/

[29] Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, Roger Chou, Julie Glanville, Jeremy M Grimshaw, Asbjørn Hróbjartsson, Manoj M Lalu, Tianjing Li, Elizabeth W Loder, Evan Mayo-Wilson, Steve McDonald, Luke A McGuinness, Lesley A Stewart, James Thomas, Andrea C Tricco, Vivian A Welch, Penny Whiting, and David Moher. 2021. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ* (March 2021), n71. https://doi.org/10.1136/bmj.n71

[30] Olga Petrovska, Lee Clift, Faron Moller, and Rebecca Pearsall. 2024. Incorporating Generative AI into Software Development Education. In *Proceedings of the 8th Conference on Computing Education Practice* (Durham, United Kingdom) *(CEP '24)*. ACM, New York, NY, USA, 37–40. https://doi.org/10.1145/3633053.3633057

[31] Siddhartha Prasad, Ben Greenman, Tim Nelson, and Shriram Krishnamurthi. 2023. Generating Programs Trivially: Student Use of Large Language Models. In *Proceedings of the ACM Conference on Global Computing Education Vol 1* (Hyderabad, India) *(CompEd 2023)*. ACM, New York, NY, USA, 126–132. https://doi.org/10.1145/3576882.3617921

[32] James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N. Reeves, and Jaromir Savelka. 2023. The Robots Are Here: Navigating the Generative AI Revolution in Computing Education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education* (Turku, Finland) *(ITiCSE-WGR '23)*. ACM, Turku Finland, 108–159. https://doi.org/10.1145/3623762.3633499

[33] James Prather, Brent N. Reeves, Paul Denny, Brett A. Becker, Juho Leinonen, Andrew Luxton-Reilly, Garrett Powell, James Finnie-Ansley, and Eddie Antonio Santos. 2023. "It's Weird That it Knows What I Want": Usability and Interactions with Copilot for Novice Programmers. *ACM Transactions on Computer-Human Interaction* 31, 1 (Nov. 2023), 1–31. https://doi.org/10.1145/3617367

[34] Basit Qureshi. 2023. ChatGPT in Computer Science Curriculum Assessment: An analysis of Its Successes and Shortcomings. In *2023 9th International Conference on e-Society, e-Learning and e-Technologies* (Portsmouth, United Kingdom) *(ICSLT '23)*. ACM, New York, NY, USA, 7–13. https://doi.org/10.1145/3613944.3613946

[35] Parsa Rajabi, Parnian Taghipour, Diana Cukierman, and Tenzin Doleck. 2023. Exploring ChatGPT's impact on post-secondary education: A qualitative study. In *Proceedings of the 25th Western Canadian Conference on Computing Education* (Vancouver, BC, Canada) *(WCCCE '23)*. ACM, New York, NY, USA, 1–6. https://doi.org/10.1145/3593342.3593360

[36] Lianne Roest, Hieke Keuning, and Johan Jeuring. 2024. Next-Step Hint Generation for Introductory Programming Using Large Language Models. In *Proceedings of the 26th Australasian Computing Education Conference* (Sydney, NSW, Australia) *(ACE '24)*. ACM, New York, NY, USA, 144–153. https://doi.org/10.1145/3636243.3636259

[37] Sue Sentance, Jane Waite, and Maria Kallia. 2019. Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education* 29, 2-3 (July 2019), 136–176. https://doi.org/10.1080/08993408.2019.1608781

[38] Raphael Serafini, Clemens Otto, Stefan Albert Horstmann, and Alena Naiakshina. 2024. ChatGPT-Resistant Screening Instrument for Identifying Non-Programmers. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. ACM, New York, NY, USA, 1–13. https://doi.org/10.1145/3597503.3639075

[39] Judy Sheard, Paul Denny, Arto Hellas, Juho Leinonen, Lauri Malmi, and Simon. 2024. Instructor Perceptions of AI Code Generation Tools - A Multi-Institutional Interview Study. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. ACM, New York, NY, USA, 1223–1229. https://doi.org/10.1145/3626252.3630880

[40] Brad Sheese, Mark Liffiton, Jaromir Savelka, and Paul Denny. 2024. Patterns of Student Help-Seeking When Using a Large Language Model-Powered Programming Assistant. In *Proceedings of the 26th Australasian Computing Education Conference* (Sydney, NSW, Australia) *(ACE '24)*. ACM, New York, NY, USA, 49–57. https://doi.org/10.1145/3636243.3636249

[41] Abdulhadi Shoufan. 2023. Can Students without Prior Knowledge Use ChatGPT to Answer Test Questions? An Empirical Study. *ACM Transactions on Computing Education* 23, 4 (Dec. 2023), 45:1–45:29. https://doi.org/10.1145/3628162

[42] TJ Wang, DV Díaz, C Brown, Y Chen, and IEEE. 2023. Exploring the Role of AI Assistants in Computer Science Education: Methods, Implications, and Instructor Perspectives. 92–102. https://doi.org/10.1109/VL-HCC57772.2023.00018