Directions for using Generative Artificial Intelligence in Introductory Programming

MICHAEL DOLINSKY
Faculty of Mathematics and Programming Technologies,
Francisk Skorina Gomel State University,
Sovetskaya Str.104, 246019, Gomel,
BELARUS

Abstract: - This paper provides a review of the literature on the use of generative artificial intelligence (GenAI) in introductory programming education at universities. The main areas of application of generative artificial intelligence are given: specialized narrow-topic developments, integration into online learning platforms and testing solutions, students working with GenAI without restrictions, guiding systems for the interaction of students with GenAI (without providing solutions), teacher assistance, tools for developing intelligent teaching systems. The paper also contains a review of works that analyze the results achieved and unresolved problems.

Key-Words: - generative artificial intelligence, teaching programming to first-year students, intelligent teachingsystems, ChatGPT, computational thinking, review,

Received: March 18, 2024. Revised: October 17, 2024. Accepted: November 17, 2024. Published: December 31, 2024.

1 Introduction

At present, the development of digital systems, electronic devices, and digital signal processing tools is to a significant extent tied to the creation of software. In turn, training programmers from the entry-level causes enormous difficulties. The goal of initial programming training at the university is to develop the skills of "Computational Thinking" [1], which includes the following components: abstraction, decomposition, pattern recognition, algorithmization, and debugging. A significant increase in learning efficiency comes from the use of online platforms for teaching programming, such as EduCoder [2], [3], HTProgramming [4] and Code4brownies [5].

A new stage in increasing the efficiency of initial programming training is associated with the introduction of generative artificial intelligence. Text-based generative systems have a chatbot as the front end to interact with the user and are based on LLMs - generative models that can produce new content based on the data they are trained on. The most famous LLMs: are GPT (OpenAI), Gemini (Google), Llama (Meta), and Claude3 (Anthropic). User interaction with a chatbot is implemented by requests. [6] formulated tasks for such chatbots during the initial programming learning process: explaining basic knowledge, constructing code, explaining code, refactoring code, formatting code, checking coding style, and commenting on code. Standard practices for using GenAI in teaching programming are presented in [7].

Designing and formulating queries in order to obtain the optimal answer is often called prompt engineering. An excellent illustration of how to obtain fantastic results using prompt engineering is presented in [8], dedicated to contextual additional training of ChatGPT in the field of bioinformatics. The GeneGPT presented in this work uses the NCBI Web API documentation and in-query demonstration for contextual learning. Then GeneGPT answers GeneTuring and GeneHop tests using the NCBI Web API.

To complete the picture, we must also take into account the opinion of skeptics who believe that at the initial stage of learning programming, a complete ban on the use of GenAI by students is necessary [9].

2 Problem Formulation

Initial programming training at universities faces enormous difficulties and, as a result, a high dropout rate and insufficient training even for those who pass the exam. Currently, improving the quality of initial

programming training is largely associated with the introduction of generative artificial intelligence into the learning process. The main directions of such introduction are discussed further in this paper.

3 Problem Solution

3.1 Specialized Narrow-Topic Developments

[10] presents a product that helps write Python programs, formatted according to the PEP-8 standard. MMAPR [11] allows you to automatically correct errors in student programs written in Python. Codeboard [12] helps students find and correct errors in the proposed JavaScript programs through dialogue. [13] presents a product that provides automatic control of compliance with rules for Java programs posted by students on GitHub. [14] describes a program that helps students carry out tasks on SQL queries to PostgreSQL in a dialogue. [15] and [16] presented developments that help students complete programming tasks in dialogue, answering their questions and offering explanatory examples.

3.2 Platforms that Allow Students to Transparently Interact with ChatGPT

It seems more effective to integrate GenAI into online systems for training and testing programs. The following systems offer this approach.

TutorBot+ [17], [18] is integrated into a WEB training system and an online program testing platform; it allows you to take the conditions of problems, obtain solutions in dialogue with GenAI, and check solutions.

KOGI [19] is a learning support system that integrates ChatGPT and the Jupyter framework. KOGI helps the student get advice from ChatGPT in response to errors and questions. KOGI is implemented in two introductory courses: Algorithms, and Data Science. As a result, there is a significant reduction in the number of unresolved student errors and teachers receive information about questions and errors.

IPSSC - Intelligent programming scaffolding system using ChatGPT, [20]. Instead of having the student interact directly with ChatGPT, the authors developed three modules: Solution Assessment (SA), Code Assessment (CA), and Free Interaction (FI). In the SA module, students decompose a complex problem into several simple ones and abstract them so that the solution is suitable for similar problems (thus, this module focuses on decomposition and abstraction). In the CA module, students design algorithms, write code, and continuously debug code, using the module to identify errors in the code, and then fix them. ChatGPT helps improve algorithms and code. In the FI module, students can interact with ChatGPT directly, discussing topics that are not covered in the SA and CA modules.

Coding Step [21] is a web application designed to teach the basics of Python programming. It is a beginner-friendly programming environment with a series of exercises introducing new concepts. The student can move through these tasks independently with access to the GenAI code generator from OpenAI Codex. Users enter text describing the behavior of the program in natural language, and by clicking on the “Generate Code” button they receive a program generated by OpenAI Codex. Each Codex API call is customized by adding predefined examples, existing code in the editor, and user-requested behavior. This ensures that the GenAI code generator will generate Python code for the novice programmer that matches the user's context. Students work with the generated code. This approach reduces the number of errors and reduces the time to complete the task.

LearningProgrammingWithGPT [22] is an environment that is used by students in the CS1 course when learning Python. The course integrates ChatGPT as a means to support students and instructors in the classroom. The platform acts as a medium between students and their GenAI instructor, providing personalized educational material based on pre-prompts built into student requests. The training program itself consists of three parts:

1) Introduction – Variables – Operators – Input and Output
2) Comments – Conditions – Arrays (Lists) – Errors and Debugging
3) Loops – Functions – Library

GPTutor [23] is an extension for VS Code that supports more than 50 programming languages, including C, C++, Java, Python, and C#. GPTutor allows users to customize tooltips for different programming languages and scripts. It provides code explanations, reviews, and customizable tooltips. The Python course (8 weeks) includes assignments for a week on the topic of the week (students can use ChatGPT); homework (students can discuss difficult

issues with ChatGPT); individual mini-project in the last week - a unique program of less than 1000 lines of code, on any topic from games to chatbots, you can use ChatGPT, but include the ChatGPT dialogue in the report. ChatGPT can be used not only for debugging, explanation, and optimization, but also for brainstorming ideas, clarifying project requirements, and gaining a deeper understanding of the technical aspects of the project.

Copilot [24] is a GenAI assistant that uses LLM trained on billions of lines of code and is available as a free IDE plugin for students. Copilot can generate correct and readable code in common programming languages.

[25] additionally provides a check for plagiarism.

3.3 Help without Providing Solutions

As part of a constructive response to skeptics GenAI in initial programming education products are being developed that monitor and modify student requests and GenAI responses in such a way as to provide assistance without presenting solutions.

The developers of CodeHelp [26], [27] claim that programming teachers have huge problems being able to devote enough time to each student. LLM models help students solve their problems, but there are two types of dangers: solving problems instead of students, and solving problems incorrectly. CodeHelp provides students with on-demand help without offering immediate solutions, an example of a "Socratic" tutor who avoids revealing solutions directly to the user. CodeHelp supports any programming language because it uses GenAI. At the same time, the authors emphasize that CodeHelp is not a replacement for a human teacher, but an addition to it. A student can ask for help at any time and immediately receives a response - advice without providing a complete solution, which is obtained by queries to the AI based on student input. In addition, the teacher has the opportunity to monitor student work. CodeHelp is designed to answer any student's questions without generating a solution code, regardless of the wording of the request. This is achieved by a series of different queries that include instructions not to generate code, and separately verify that the code is not present in the response from the LLM. The goal is to ensure that students receive step-by-step help, as a mixture of natural language and pseudocode, students must write the correct code themselves. CodeHelp also uses prompts to generate “positive and approving” responses, just like a “human teacher” does. CodeHelp developers emphasize three main characteristics of their digital teacher assistant: teach, rather than give ready-made solutions, conduct effective dialogue, and take into account the student’s level of preparation.

CodeAid [28] also avoids direct answers with code, trying to guide the student to solve the problem. CodeAid allows students to: ask a general programming question, ask a question about the presented code, ask for an explanation of the presented code, help fix a bug in the presented code, and help write the code. During the semester of testing, the following data was collected that allowed the authors to improve CodeAid: more than 8,000 student interactions with CodeAid and their evaluation of the generated response, 10 weekly student reports on the use of CodeAid and other resources, 22 student interviews about CodeAid, a final anonymous review comparing CodeAid with ChatGPT.

The developers of NotebookGPT [29] believe that direct use of ChatGPT can interfere with student learning, so NotebookGPT gives access to GPT but does not return complete solutions, providing: feedback on programming style, explanations of how pieces of code work, help with debugging code, the ability to see alternative solutions to problems.

AI TA [30] helps students perform decomposition - that is, partitioning tasks into subtasks. According to AI TA developers, models such as ChatGPT, LLaMA, and Gemini in the context of teaching programming show remarkable abilities to understand, generate, and explain code, making them strong candidates for Teacher Assistants in programming courses. They can correct errors in the code, explain why errors occurred, and discuss possible approaches to solving various programming problems. However, there are also problems - they can generate erroneous code and even different answers to the same question. That's why AI TA supports “Subgoal learning” an effective learning strategy that helps students break down complex problems into subtasks.

StAP-tutor [31] provides recommendations of the next step to solve the problem. Experiments have shown that in most cases StAP tutor recommends the correct step, and it is personalized to the student's code and approach. However, there are also cases of erroneous advice.

TeachYou+AlgoBo [32] offers an LBT (learning by teaching - teaching yourself by teaching

someone). Here TeachYou is an LBT environment for teaching algorithms, and AlgoBo is a learning chatbot for problem solving. Instead of designing learning agents from scratch, the authors propose a top-down methodology in which they use different LLMs to simulate learners. In their formative study, the authors asked 15 novices in programming to perform LBT with ChatGPT, who were promptly asked to act as learners. The authors found that it is necessary to: limit the level of knowledge of the LLM agent, agent-initiated questions "why" and "how"; feedback on teaching and learning methods. As a result, the author's approach assumes that in TeachYou, the student solves an algorithm programming problem (for example, binary search) and then reflects on this work by teaching AlgoBo.

3.4 Platforms for the Development of Intelligent Teaching Systems

Further increases in efficiency are associated with the development of platforms for creating GenAI tutors, [33].

VoLL-KI [34] is a research project with the goal of creating a set of tools at different levels of abstraction to improve the learning process in universities using artificial intelligence. This means tools such as training planning; recommendation systems for services for answering student questions and selecting educational materials; training using virtual reality; educational chatbots; adaptive learning environments;

The practical implementation of the Apprentice Tutor Builder (ATB) platform for creating intelligent teaching systems (ITS) is described in [35]. ITS is an educational technology that provides students with course content, individually proposed practical tasks, and effective feedback. According to the authors of ATB, ITS are very effective, but their development is complex, time-consuming, and requires special programming and teaching knowledge, which hinders their widespread use. That's why the authors are developing ATB, a platform that simplifies the creation and personalization of ITS tutors. A teacher can use drag-and-drop technology to create a tutor interface, and then interactively train a GenAI agent to solve problems. Learning is achieved through repeated interactive interactions, including demonstrations, feedback, and custom labels. As a result, you need to spend 200-300 hours on one hour of training. 14 instructors appreciated the effectiveness of using ATB but indicated desired opportunities to reduce development time. As a result of the analysis of this test work, the authors created many recommendations for using the ATB platform and the GenAI agent.

3.5 Help the Teacher

A separate direction for GenAI is to help for teachers. Here we can note such products as Robosourcing [11] for generating practical tasks in a scalable manner, VizGroup [36], providing control over student activities in large classes (100+ students), checking student solutions using ChatGPT [37], PAIGE [38] is a fraud detection system that identifies texts written by GenAI rather than by humans.

3.6 Reflective Works

[39] proposes ways to use ChatGPT for self-learning. [40], [41], [42] describe the advantages and disadvantages of using GenAI for teaching programming. [43] presents the results of a survey of students about the benefits and problems of using GenAI to teach programming. The collection and analysis of student chat protocols in an introductory programming course is proposed in [44]. A technology for testing ChatGPT's ability to answer programming questions was proposed in [45]. [46] offers the experience of "How to teach solving problems using examples". [47] proposes to solve the problems of the progress of artificial intelligence in initial programming teaching, using approaches proposed in teaching mathematics after the invention of calculators and computers.

4 Conclusion

To summarize, the capabilities of GenAI described in the article can be divided functionally into two groups:

- How students can use GenAI: automatic syntax correction, completing assignments, obtaining multiple solutions to one problem, receiving explanations for solutions and error messages, and in the future, getting a virtual teacher.
- How teachers can use GenAI: generation of programming assignments, generation of new exercises based on existing ones, contextualization of task conditions, and evaluation of student work.

Options for future programming training with GenAI code generation: Ideally, change tasks so that students can use code generation and at the same time gain their own problem-solving skills.

Radicals propose a shift in focus in programming courses. AI code generators can solve low-level implementation problems, and students will develop high-level algorithms. Because software developers in the future will be more focused on project-oriented roles, the focus of courses should change to developing AI code generator queries, code evaluation, and debugging.

Rationalists suggest teaching beginners using traditional approaches and teaching AI code generation in subsequent courses.

References:

[1] Kaleem M., Hassa M. A.n, Khurshid S. K. A Machine Learning-Based Adaptive Feedback System to Enhance Programming Skill Using Computational Thinking, *IEEE Access*. 2024. Vol. 12. P. 59431–59440. doi: 10.1109/access.2024.3391873.

[2] Zhang S., Yang J., Sang X. Exploring the Applications of EduCoder Platform in Blended Teaching for Computer Major, *Journal of Education and Educational Research*. 2023. Vol. 4, № 2. P. 100–103. doi: 10.54097/jeer.v4i2.10819.

[3] Li Z., Zhang S., Sang X. Exploration of Machine Learning Teaching based on the EduCoder Platform , *Journal of Education and Educational Research*. 2023. Vol. 4, № 3. P. 130–133, [Online]. https://drpress.org/ojs/index.php/jeer/article/view/10819/10528 (Accessed Date: January 4, 2025).

[4] Figueiredo J., Garcia-Penalvo F. Teaching and Learning Tools for Introductory Programming in University Courses, *2021 International Symposium on Computers in Education (SIIE)*. 2021. doi:10.1109/siie53363.2021.9583623 https://doi.org/10.1109/SIIE53363.2021.9583623.

[5] Phan V., Hicks E. Code4Brownies: an active learning solution for teaching programming and problem-solving in the classroom, *Proc. of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. 2018. P. 153–158. doi: 10.1145/3197091.3197128.

[6] Herden O. Integration of Chatbots for Generating Code into Introductory Programming Courses, *Proc. of International Conference "Future of education.* 2024, [Online]. https://conference.pixel-online.net/files/foe/ed0014/FP/9091-ICT6673-FP-FOE14.pdf (Accessed Date: January 4, 2025).

[7] Koppe Ch., Keuning H., Lykourentzou I., Chacon I. A., Sosnovsky S. *Practices for the Application of Generative AI in Programming Education.* Research project, [Online]. https://www.uu.nl/sites/default/files/GenAI%20in%20ProgEd%20practices.pdf (Accessed Date: January 4, 2025).

[8] Jin Q., Yang Y., Chen Q., Lu Z. GeneGPT: augmenting large language models with domain tools for improved access to biomedical information, *Bioinformatics*. 2024. Vol. 40, № 2. P. btae075. doi: 10.1093/bioinformatics/btae075.

[9] Lau S., Guo P. 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, *Proc. of the 2023 ACM Conf. on Int. Computing Education Research*. 2023. Vol. P. 106–121. doi: 10.1145/3568813.3600138.

[10] Farah J. C., Spaenlehauer B., Ingram S., Purohit A. K., Holzer A., Gillet D. Harnessing Rule-Based Chatbots to Support Teaching Python Programming Best Practices, *Lecture Notes in Networks and Systems*. 2024. P. 455–466. doi: 10.1007/978-3-031-51979-6_47.

[11] Cambaz D. *Use of AI-driven code generation models in teaching and learning programming: a systematic literature review*. Bachelor thesis, TU Delft, Faculty Electrical Engineering, Mathematics and Computer Science, Delft, The Netherlands, 2023, [Online]. https://repository.tudelft.nl/islandora/object/uuid:4071531b-2dd0-4001-b67c-6351761d4821 (Accessed Date: January 6, 2025).

[12] Roca M. D. L., Chan M. M., Garcia-Cabot A., Garcia-Lopez E., Amado-Salvatierra H. The impact of a chatbot working as an assistant in a course for supporting student learning and

engagement, *Computer Applications in Engineering Education.* 2024. P. e22750. doi: 10.1002/cae.22750.

[13] Bobadilla S., Glassey R., Bergel A., Monperrus M. SOBO: A Feedback Bot to Nudge Code Quality in Programming Courses, [Online]. https://arxiv.org/pdf/2303.07187 (Accessed Date: January 6, 2025).

[14] Sengewald J., Wilz M., Lackes R. AI-Assisted Learning Feedback: Should Gen-AI Feedback Be Restricted to Improve Learning Success? A Pilot Study in a SQL Lecture, *Proc. of ECIS* 2024. 2024. p.12, [Online]. https://aisel.aisnet.org/ecis2024/track13_learning_teach/track13_learning_teach/12 (Accessed Date: January 6, 2025).

[15] Pahi K., Hawlader S., Hicks E., Zaman A., Phan V. Enhancing active learning through collaboration between human teachers and generative AI, *Computers and Education Open.* 2024. Vol. 6. P. 100183. doi: 10.1016/j.caeo.2024.100183.

[16] Roldan-Alvarez D., Mesa F. J. Intelligent Deep-Learning Tutoring System to Assist Instructors in Programming Courses, *IEEE Transactions on Education.* 2024. Vol. 67, № 1. p.153–161. https://doi.org/10.1109/TE.2023.3331055.

[17] Martinez-Araneda C., Gutierrez M., Maldonado D., Gomez P., Segura A., Vidal-Castro C. Designing a Chabot to support problem-solving in a programming course, *Proc. of 18th International Technology, Education and Development Conference.* 2024. p. 966–975. doi: 10.21125/inted.2024.0317.

[18] Martınez-Araneda C., Gutierrez Valenzuela M., Gomez Meneses P., Maldonado Montiel D., Segura Navarrete A., Vidal-Castro C. How Useful TutorBot+ is for Teaching and Learning in Programming Courses: a Preliminary Study, *Proc. of 42nd IEEE International Conference of the Chilean Computer Science Society (SCCC).* 2023. P. 1–7. doi: 10.1109/sccc59417.2023.10315697.

[19] Kuramitsu K., Obara Y., Sato M., Obara M. KOGI: A Seamless Integration of ChatGPT into Jupyter Environments for Programming Education, *Proc. of the 2023 ACM SIGPLAN Int. Symposium on SPLASH-E.* 2023. P. 50–59. doi: 10.1145/3622780.3623648.

[20] Liao, J., Zhong, L.(2024). Scaffolding Computational Thinking with ChatGPT. *IEEE Transactions on Learning Technologies,* Vol. 17, 2024.

[21] Sarshartehrani F., Mohammadrezaei E., Behravan M., Gracanin D. Enhancing E-Learning Experience Through Embodied AI Tutors in Immersive Virtual Environments: A Multifaceted Approach for Personalized Educational Adaptation, *Lecture Notes in Computer Science.* 2024. P. 272–287. https://doi.org/10.1007/978-3-031-60609-0_20.

[22] Abolnejadian M., Alipour S., Taeb K. Leveraging ChatGPT for Adaptive Learning through Personalized Prompt-based Instruction: A CS1 Education Case Study, *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems.* 2024. p.1–8. doi:10.1145/3613905.3637148.

[23] Ma B., Li C., Shin'ichi K. Enhancing programming education with ChatGPT: a case study on student perceptions and interactions in a Python course, [Online]. https://arxiv.org/pdf/2403.15472v3 (Accessed Date: January 6, 2025).

[24] Menon P. Exploring GitHub Copilot assistance for working with classes in a programming course, *Issues in Information Systems.* Vol. 24, № 4p. 66, 2023; doi: 10.48009/4_iis_2023_106.

[25] Hellas A., Leinonen J., Leppanen L. Experiences from Integrating Large Language Model Chatbots into the Classroom , arxiv.org. 2023, [Online]. https://arxiv.org/pdf/2406.04817 (Accessed Date: January 6, 2025).

[26] Liffiton M., Sheese B., Savelka J., Denny P. CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes, arxiv.org. 2023, [Online]. https://arxiv.org/pdf/2308.06921 (Accessed Date: January 6, 2025).

[27] Denny P., MacNeil S., Savelka J., Porter L., Luxton-Reilly A. Desirable Characteristics for AI Teaching Assistants in Programming Education, arxiv.org. 2024, [Online]. https://arxiv.org/pdf/2405.14178v1 (Accessed Date: January 6, 2025).

[28] Kazemitabaar M. Ye R., Wang X., Henley A.Z., Denny P., Craig M., Grossman,T. CodeAid: Evaluating a Classroom Deployment

of an LLM-based Programming Assistant that Balances Student and Educator Needs, arxiv.org. 2024, [Online]. https://arxiv.org/pdf/2401.11314 (Accessed Date: January 6, 2025).

[29] George S. D., Dewan P. NotebookGPT - Facilitating and Monitoring Explicit Lightweight Student GPT Help Requests During Programming Exercises, *Companion Proceedings of the 29th International Conference on Intelligent User Interfaces.* 2024. P. 62–65. doi: 10.1145/3640544.3645234.

[30] Lee Ch., Myung J., Han J., Jin J. Alice Oh Learning from Teaching Assistants to Program with Subgoals: Exploring the Potential for AI Teaching Assistants, arxiv.org. 2023, [Online]. https://arxiv.org/pdf/2309.10419 (Accessed Date: January 6, 2025).

[31] Roest L., Keuning H., Jeuring J. Next-Step Hint Generation for Introductory Programming Using Large Language Models, *Proc. of the 26th Australasian Computing Education Conference.* 2024. p.144–153. doi: 10.1145/3636243.3636259.

[32] Jin H., Lee S., Shin H., Kim J. Teach AI How to Code: Using Large Language Models as Teachable Agents for Programming Education, arxiv.org. 2023, [Online]. https://arxiv.org/pdf/2309.14534v2 (Accessed Date: January 6, 2025).

[33] Keshtkar F., Rastogi N., Chalarca S., S. A. C. Bukhari. AI Tutor: Student's Perceptions and Expectations of AI-Driven Tutoring Systems: A Survey-Based Investigation, *Proc. of FLAIRS-37.* 2024, [Online]. https://doi.org/10.32473/flairs.37.1.135314 (Accessed Date: January 6, 2025).

[34] Kohlhase , Berges M., Grubert J., Henrich A., Landes D., Leidner J.L., Mittag F., Nicklas D., Schmid U., Sedlmaier Y., vom Ende A.U., Wolter D. Project volL-KI, KI - Kunstliche Intelligenz. 2024. doi: 10.1007/s13218-024-00846-9.

[35] Smith G., Gupta A., MacLellan Ch. Apprentice Tutor Builder: A Platform for Users to Create and Personalize Intelligent Tutors, arxiv.org. 2024, [Online]. https://arxiv.org/pdf/2404.07883 (Accessed Date: January 6, 2025).

[36] Tang X. , Wong S., Pu K., Chen X., Yang Y., Chen Y. VizGroup: An AI-Assisted Event-Driven System for Real-Time Collaborative Programming Learning Analytics, arxiv.org. 2024, [Online]. https://arxiv.org/pdf/2404.08743v1 (Accessed Date: January 6, 2025).

[37] Estevez-Ayres I., Callejo P., Hombrados-Herrera M. ´ A., Alario-Hoyos C., Delgado Kloos C. ´ Evaluation of LLM Tools for Feedback Generation in a Course on Concurrent Programming, *Int. Journal of Artificial Intelligence in Education.* 2024. doi: 10.1007/s40593-024-00406-0.

[38] Shanto S. S., Ahmed Z., Jony A. I. PAIGE: A generative AI-based framework for promoting assignment integrity in higher education, *STEM Education.* 2023. Vol. 3, № 4. p.288-305. https://doi.org/10.3934/steme.2023018.

[39] Silva R. Using ChatGPT as a Virtual Tutor for Teaching Algorithms. [Technical Report], Department of Computer Science, Federal University of Juiz de Fora, Juiz de Fora and Governador Valadares, Brazil, 2024, [Online]. https://www.researchgate.net/publication/380244926_Using_ChatGPT_as_a_Virtual_Tutor_for_Teaching_Algorithms (Accessed Date: January 6, 2025).

[40] Perera K. G. D. K., Wijayanayake J., Prasadika J. Factors Affecting the Effectiveness of Generative Artificial Intelligence Apps on University Students' Programming Language Learning in Sri Lanka: A Systematic Literature Review, *2024 4th Int. Conf. on Advanced Research in Computing (ICARC).* 2024. P. 276–281. doi: 10.1109/icarc61713.2024.10499744.

[41] Liu S., Grey B., Watkins R. Assessing Risks, Challenges and Opportunities of Generative AI in Computer Programming Education, *Lightning Talk Journal of Computing Sciences in Colleges.* 2024. Vol. 39, № 8. P. 210-211.

[42] Lin C.-C., Huang A. Y. Q., Lu O. H. T. Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review, *Smart Learning Environments.* 2023. Vol. 10, № 1. https://doi.org/10.1186/s40561-023-00260-y.

[43] Le T. M. T. *Utilizing AI Tools in Learning Introductory Programming: An Analysis of*

Tampere University Students' Strategies and Outcomes. Bachelor Thesis. Tampere University Science and Engineering, Tampere, Finland, 2024.

[44] Scholl A., Schiffner D., Kiesler N. Analyzing Chat Protocols of Novice Programmers Solving Introductory Programming Tasks with ChatGPT, arxiv.org. 2024, [Online]. https://arxiv.org/pdf/2405.19132 (Accessed Date: January 6, 2025).

[45] Leidner J. L., Reiche M. Language-Model Assisted Learning How to Program?, *Artificial Intelligence. ECAI 2023 International Workshops.* 2024. P. 425–438. doi: 10.1007/978-3-031-50485-3_41.

[46] Hocquettea C., Langerb J., Croppera A., Schmidb U. Can Humans Teach Machines to Code?, arxiv.org. 2024, [Online]. https://arxiv.org/pdf/2404.19397 Last Accessed Date: 31-Jan-2025

[47] Jacques L. Teaching CS-101 at the Dawn of ChatGPT, ACM Inroads. 2023. Vol. 14, № 2. p.40-46. https://doi.org/10.1145/3595634.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The author contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The author has no conflicts of interest to declare.