Information Fusion Research (2024) Volume 2 Issue 2
doi: 10.59429/ifr.v2i2.8881
ISSN: 3029-2220 (O)

RESEARCH ARTICLE

Use of generative artificial intelligence software in the process of introductory programming training at university

Michael Dolinsky

Ph.D, docent, docent of the chair "Department of Mathematical Problems of Control and Informatics"
Gomel State University named after F. Skorina, Gomel, Belarus, Sovietskaya str. 104, 246023, Gomel

***Corresponding author:** Michael Dolinsky, dolinsky@gsu.by

ABSTRACT

This paper provides a review of the literature on introductory programming training in universities and the use of generative artificial intelligence software (GAIS) for these purposes. During introductory programming training in universities, the tasks of developing five skills are highlighted: abstraction (determining what information about an entity/object needs to be preserved and what to ignore); generalization (formulating a solution in general terms so that it can be applied to solve various problems); decomposition (breaking complex problems into smaller subtasks that are easier to understand and solve); algorithmization (determining step by step a set of operations/actions in order to solve a problem); debugging (localization and elimination of errors). There are also many programming languages used in introductory education at universities, such as Pascal, C++, Python, C#, Java. A review of the literature on the use of GAIS in teaching of programming at universities indicated the following main problems and tasks: to prevent the use of GAIS to obtain undeserved grades in the subject; develop techniques and examples of using GAIS in the educational process to personalize, increase intensity and quality; explore new areas of using GAIS, for example, developing skills in using GAIS to solve real production problems (GAIS operator). A description is given of the author's experience of using GAIS for introductory programming training for first-year students of the Faculty of Mathematics and Programming Technologies of Gomel State University named after F. Skorina. The main futures of authors approach are the following: automatic personalized training, low entry threshold, multiple entry points, personalization, intensification, freedom to choose a program development language (in current practice between five programming languages: Pascal, Python, Java, C#, C++. If the need arises, new languages can be added).

Keywords: generative artificial intelligence; programming training for freshmen; DL.GSU.BY

1. Introduction

Beginner programming courses. Subject "programming" is basic for the first years of all university engineering majors and very difficult for most students[1]. A comparative analysis of approaches to teaching programming in introductory courses is given in[2]. A comparative analysis of the state of teaching five basic concepts in introductory programming courses (abstraction, decomposition, algorithmization, debugging and

ARTICLE INFO
Received: 5 October 2024 |Accepted: 17 November 2024 |Available online: 27 November 2024

CITATION
M. Dolinsky. Use of generative artificial intelligence software in the process of introductory programming training at university. *Information Fusion Research* 2024; 2(2): 8881.doi: 10.59429/ifr.v2i2.8881

problem solving) in European countries, North America, Oceania, Asia, Latin America and Africa are given in [3]. In 2020, IEEE and ACM, two leading associations of computer and computer science professionals, created Computing Curricula, which defined the strategy for teaching computer science, including introductory programming courses[4].

Among the means offered by experts to increase the effectiveness of teaching programming in the first year of university are:

- visual programming environments[5, 6]
- use of specially developed WEB platforms[7-9]
- application of a project-oriented approach[10,11]
- special teaching aids for teaching programming[12]
- original methods of teaching programming[13], including modern approaches to teaching, modified pedagogy, creating an attractive and realistic programming learning environment[14]
- system for automatic evaluation of student solutions[15]
- gamification[16,17]
- a special software tool for monitoring students' progress during the semester[18].

Impact of artificial intelligence programs on learning. It is noted in[19] that AI programs such as ChatGPT are able to perform tasks in introductory programming courses instead of students, because ChatGPT understands not only text descriptions of tasks, but also programs written in various programming languages such as Python, Java, C++, etc. Accordingly, teachers are faced with the task of changing the relevant assignments. [20] presents one approach to teaching introductory programming courses using artificial intelligence (AI). Among other things, it is proposed to give tasks for the use of AI generating programs for various task representations: textual representations (algorithms written in a programming language or pseudocode), visual representations (for example, UML diagrams), numerical and formulaic representations. Another option for tasks is the generation of several solutions to the same problem, followed by a comparative analysis of implementations. The third is the student's explanation of the AI-generated solutions.[21] propose using ChatCPT to evaluate student-developed programs, noting that historically this has been done either by manual review or automated testing against tests. Traditional approaches to teaching programming focus on the practice of writing code. But with the advent of AI code generators, it is necessary to change these approaches towards developing the skills of forming effective text queries to AI code generators[22,23] describes the use of ChatGPT in data processing, citing the advantages of this approach as personalized learning, explanations of concepts, generation of code with explanations, and evaluation of work. Data science includes three essential required courses: databases, data analytics, machine learning. ChatBot, which helps students master the design of Python programs in accordance with the PEP-8 standard, is presented in[24].

Analysis of all these leads to the following conclusions:

1. The use of generative artificial intelligence programs is inevitable and can significantly improve the quality of the educational process in a wide variety of and as yet unexpected ways.

2. Currently, one of the most important tasks is to protect against the use of GAIS by unscrupulous students for laboratory and practical work.

2. The purpose of the semester course on introductory prpgramming

The author has been teaching introductory programming courses to first-year students for many years[25,26]. First year student at the Faculty of Mathematics and Programming Technologies of Gomel State University. F. Skaryna enrolls students who, for the most part, do not know how to write programs, at the same time there are first-year students who began independent study even before entering the university, making their own choice of programming language (in the 2023-2024 academic year this is one of 5 languages: Pascal, C++, Python, Java, C#).

The purpose of the semester-long course on programming and the basics of algorithmization (a total of 10 academic hours per week) is to lay the foundations of basic theoretical knowledge and practical skills in the following areas:

Based on the textual statement of the problem, make a mathematical/algorithmic statement.

The source text of the task can contain from several lines to several pages. The student must be able to separate useful information from irrelevant information and, at the end of this process, reformulate the problem statement in the form of a mathematical or algorithmic formulation, using such concepts as numbers, symbols, strings, one-dimensional and multidimensional arrays of numbers, symbols and strings as the intended objects of processing and results.

Develop an algorithm

The development of an algorithm involves a textual representation of a sequence of actions leading to obtaining the result required by the formulation of the problem for the given initial data. The algorithm must be described in such a way that it can be used by any student of the course to obtain the correct result from the proposed input data.

Write a program

Most students are taught in the Pascal programming language for the following reasons:

- Pascal was designed as language for teaching programming, and therefore is easier than others to learn by students who do not have any prior training.
- Since 1999, under the leadership of the author, the distance learning system DL.GSU.BY has been developed [27,28], which has built an effective environment for automated personalized training in programming in Pascal “from scratch” for schoolchildren of various age ranges: 1-4 grades, 5- 8th grade, 9-11th grade. The use of this environment, despite the strong

differentiation in the preliminary preparation of first-year students, allows us to ensure the practical implementation of the main paradigm of the author's teaching "every student works every minute at every lesson." The student is always offered a feasible task, and he himself can regulate the level of increase in complexity of the tasks proposed for completion. In this case, it is appropriate to give an analogy with many stairs leading to the same height, but the steps that vary greatly can be very small, larger, even larger, even very large.

- For students, there are notes of reference diagrams of lectures prepared by the author on the topics "Introduction to Programming", "One-Dimensional Array - Standard Algorithms", "One-Dimensional Array - Technology for the Development of Non-Standard Algorithms", "Two-Dimensional Array", "Geometry", "Strings", "Sorting", "User-defined functions and procedures", "Queue", which are illustrated with fragments of programs in Pascal.
- The minimum subset of the syntax of the Pascal programming language is studied, such as: built-in data types (character, string, integer and real number), one-dimensional and two-dimensional arrays of built-in types, assignment operators, conditions and FOR and WHILE loops. Since the main goal is not to learn the Pascal programming language, but to develop algorithmization and programming skills.

Training in other programming languages (C++, Python, C#, Java) is also supported: there are notes of reference lecture schemes for these languages, completed and developed by interested students, automatic verification of solutions in all these programming languages is carried out, a forum is supported where any student can ask a question and receive an answer within 24 hours from a more qualified classmate or senior student.

Submit for automatic review

An important feature of the built system of personalized learning is that the conditions of the task are presented automatically; a student, having written a solution, can send it for verification and find out the result within a few seconds. For automatic verification, the solution is launched on each of the input data prepared by the author of the task and the answers of the student and the author of the task are compared; in case of ambiguity, the answers of the student's solution are checked by special programs.

Assignment of tests

A huge help for beginners in learning programming is to provide them with tests: that is, input data on which their program produced an incorrect answer along with the author's answer. This makes it quicker and easier for students to understand exactly when their program is not working correctly. The last test of the task is not given to students, which allows, on the one hand, to develop students' skills in independently developing tests, and on the other hand, ensures the impossibility of writing "fraudulent" solutions like "if the input is this, the answer is this."

Localize the error and fix it.

This is the most difficult work and, according to the author, the corresponding skills are achieved through repeated exercises in localizing and eliminating errors.

Note that all this is the basis for further training in languages and programming technologies in senior courses.

3. An assessment system that excludes the possibility of using gais for undeserved increase in grades

The author's assessment system is based on an automatically generated grade sheet, which includes in separate columns all types of student activities during the academic semester, including theory control, practice control, training, technical minimum, individual assignments, and absences from classes. Together they actually form an assessment of the student's attitude towards the educational process. The multipliers of the values in the columns are selected in such a way that all students who conscientiously apply themselves to their studies receive high grades - up to 10 and above. Another important assessment is the control section assessment. The grade for a subject is based on a minimum of these two grades ("Attitude" and "Skill"). Let me remind you that Belarus has a 10-point grading system.

A control section is a lesson in which the student is presented with 10 individual tasks for 1.5 hours, randomly selected from the available bank of tasks solved by freshmen of previous years. The score is equal to the number of completely solved problems in these 1.5 hours. Thus, for 10 different tasks (hence, 10 times), the student must complete everything he studied in the semester: formulate for himself a formal statement of the problem using the original text, develop an algorithm, write a program in the programming language of his choice, and in case of errors, find and fix them. In this case, the first 5 problems are selected from the "Technical Minimum" tasks, which students can solve during the semester. The technical minimum contains tasks on the topics: "Introduction to programming", "One-dimensional array", "Two-dimensional array", "Geometry", "Strings". Thus, a grade of 5 is guaranteed to everyone who studies conscientiously during the semester, regardless of the level of preliminary preparation before entering the university. Then follow 4 tasks on the topics "One-dimensional array", "Two-dimensional array", "Geometry", "Strings". Here, assignments are taken from assignment banks that are not available to students during the semester. The last 10th assignment on the topic "Queue" is taken from the bank of assignments available to students during the semester.

From the middle of the semester, when all the necessary basic theory has been studied, each student has the right to write a test section weekly. In this case, the estimate for the control section changes only in the direction of increase.

It is important to note that the situation is technically ensured when a student must solve problems based only on his own knowledge. Access to the Internet is blocked, as well as to all drives and folders on the local network, except for a special folder from which all files are automatically deleted when a student logs into the

system. The teacher carries out visual control so that students do not use notes and phones. Another important feature of the training system is the need to consistently solve hundreds and thousands of small problems in the learning process, which, on the one hand, repeatedly trains the necessary skills, and on the other, makes it irrational to turn to the GAIS for those students who can solve the problem themselves.

4. The results and discussion

4.1. Factors to increase students' motivation for quality learning

The following factors increase students' motivation for quality learning:

Automatic personalized training

For each theory topic, problems are organized into a tree structure, with a set of "core" problems at the trunk of the tree. If it is impossible to solve the problem, the student can click the "I don't know" button and get to the subtree of learning tasks, each of which can have its own tree of leading tasks.

Low entry threshold

In fact, no preliminary preparation is required from the student; there is even a package of tasks that allows you to learn the words program, var, longint, begin, readln, writeln, end (including the location of the corresponding Latin letters on the keyboard) and their translation, which are used when writing programs in Pascal.

Multiple entry points

The student can begin training in each of the topics either in the "Technical Minimum" section, where the simplest tasks are located, or in the "Training" section, where more complex tasks are located, or in the "Individual tasks" section, where for each topic there is a set of tasks, from where the student can choose an interesting and feasible task.

Personalization

In fact, the education system is focused on each student choosing the optimal learning route for themselves. In the "Technical Minimum" and "Training" sections, tasks are automatically proposed for solution depending on whether the previous task was solved successfully or not.

Intensification

Automatic issuance of task conditions and verification of solutions within a minute greatly intensifies the learning process, and all kinds of results tables reflecting the current achievements of students motivate many students to work better and more.

Freedom to choose a program development language

Currently, students choose in practice between five programming languages: Pascal, Python, Java, C#, C++. If the need arises, new languages can be added.

4.2. Directions for using GII in student learning

From the 2023-2024 academic year, the author invited students to actively use generative artificial intelligence programs. To do this on the system forum of DL.GSU.BY for first-year students studying programming, a special topic “Learning with the help of artificial intelligence” was created, the first message of which provides links to various programs that generate program texts, with a proposal to use them in case of the following difficult situations:

- if the solution doesn’t work, download the solution and get advice
- if you can’t come up with a solution, download the condition and receive advice and texts of solutions
- if you can’t come up with a test on which the solution does not work, upload the problem statement and solution, you get a test on which the solution does not work

It is proposed to write about the results of use in response messages in the same topic:

What did you ask, what answer did you receive, did you receive help?

There are already first responses from students, both negative and positive. Here are some fragments from them:

Positive Feedback:

“I’ve been using ChatGPT 3.5 for over a month now, it’s a very useful thing. Helps you learn independently and more. Any kind of question, even if it is very difficult to understand, can be explained many times, in a variety of ways, languages, etc. It solves complex problems poorly, but if you write a clear algorithm for a complex problem, it can be solved. It helps very well to master new topics for those who study other programming languages (C++, Java, Python, C#, etc.)

If you need an analogue summary on the main topics, and not only, write him a message like:

copied summary for PascalABC

Hello, please redo the notes for ... (your programming language), copy it into a notebook, study.

It also helps to deal with the “dead end” in a problem, find an error in the code, point out the incorrect logic of the solution and find the right path.

What I personally liked is that he answers without emotion. Even if you “don’t understand” at all, the bot tries to help as much as possible even with simple questions. A nice bonus is that the bot adapts to your skills and solves problems only using methods that you know (for this, at the beginning of the conversation, it is better to indicate what you can do, so that the bot does not use solution methods unknown to you).”

“Negative feedback”

“This bot is very far from perfect in some respects:

1) He often writes complete nonsense, because he cannot understand what he needs to do, concerns tasks like given a two-dimensional array, from each of its rows, find the maximum value and enter this value into a one-dimensional array, and then derive the minimum from the one-dimensional array (that is, tasks with 2 or more tasks).

2) The writing is incomprehensible and very tricky, the design also suffers, the code is sweeping and always with capital letters and it is also not convenient to read.

3) If the task is quite complex, then he often uses functions, and if you don't immediately tell him not to use them, then you'll end up with something incomprehensible that doesn't work.

But the bot also has advantages:

1) The bot gives good advice on the code, and its explanations are short and clear (its tips have helped me out many times.

2) The bot helps you learn, if you collaborate with the bot when working in the technical minimum and learning, then the analysis of a new topic is much easier, but for everyone, all markers are different, then it would be easier for me to work from a book, since I learned my first programming language according to the book.

ALSO: the bot is suitable for helping you study, and not for submitting tasks, because if all the tasks are very dirty with long variable names, then it is clearly a bot.

In summary: there is potential, but so far the bot is very crude, plus I haven't tested searching for errors in the code, but given that it can barely write code, then I think it will look for errors rather poorly."

The author's preliminary opinion is that generative artificial intelligence programs can greatly help in learning a programming language, especially one other than Pascal, provided the student is able to learn independently.

Generally speaking, a student can get a friend or personal teacher who is able to help him at all stages of solving a problem, including:

- based on the textual statement of the problem, make a mathematical/algorithmic statement
- develop an algorithm
- write a program based on the algorithm
- localize the error and fix it
- come up with a test on which the program does not work subject to the student's thoughtful attitude to questions and answers.

5. Conclusions and prospects for further research

This paper provides a review of the literature on introductory programming courses and the challenges

and achievements of using generative artificial intelligence programs in these courses. The following is the author's experience of teaching programming and using generative artificial intelligence programs at the Faculty of Mathematics and Programming Technologies of Gomel State University. F.Skorina. The author believes that in the future, generative artificial intelligence programs can significantly increase personalization, and, consequently, the quality of learning. However, this requires not only the development of the programs themselves, but an adjustment of methods in the direction of preparing students to study using such programs. In future the author is currently planning:

- for "problematic" tasks, collect and systematize "students' communication with GAIS" (video recording and/or screenshots of dialogue, results)
- based on the results of the analysis of the collected materials, develop methodological recommendations for working with GAIS to solve educational problems
- develop recommendations for training in the specialty "GAIS Operator".

Conflict of interest

The authors declare no conflict of interest.

References

1. Aldalur I. and Sagarna X. Improving Programming Learning in Engineering Students through Discovery Learning, IEEE Revista Iberoamericana de Tecnologias del Aprendizaje (Early Access), https://doi.org/10.1109/RITA.2023.3301409
2. Shanmugam DB, Vijayalakshmi Dr.N, Revathi N. Approaches to teaching programming: a comprehensive review and analysis, chapter in book "Research in Multidisciplinary Subjects (Volume-2) under ed. Biplab Auddy (2023)
3. Belmar HM Review on the teaching of programming and computational thinking in the world. Front. Comput. Sci. 4, 997222.https://doi.org/10.3389/fcomp.2022.997222
4. CC 2020: Computing Curricula 2020: Paradigms for Global Computing Educationhttps://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf
5. Cheng YP., Cheng SC., Yang M., Lin JM., Huang YM. Applying Computational Thinking and Formative Assessment to Enhance the Learning Performance of Students in Virtual Programming Language. In: Huang, Y. M., Rocha, T. (eds) Innovative Technologies and Learning. ICITL 2023. Lecture Notes in Computer Science, vol 14099. Springer, Cham.https://doi.org/10.1007/978-3-031-40113-8_13
6. Xu X. Teaching Reform of Virtual Instrument Technology and Application Course Based on OBE Concept. Open Access Library Journal, (2023), 10, 1-9.https://doi.org/10.4236/oalib.1110491
7. Fonseca I., Martins NC and Lopes F. A Web-based Platform and a Methodology to Teach Programming Languages in Electrical Engineering Courses," 2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE), Coimbra, Portugal , 2022, 1-6,https://doi.org/10.1109/EAEEIE54893.2022.9820594
8. Mamman M., Rdadi RS, Sadiq MA Computerized E-learning platform for teaching Python programming using Hausa language. World Journal of Advanced Engineering Technology and Sciences (2023) 09(02): 116–123https://doi.org/10.30574/wjaets.2023.9.2.0214
9. 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, 4(2)https://drpress.org/ojs/index.php/jeer/article/view/10819/10528
10. Grindei L., Constantinescu C., Bojita A., Holonec R. and Rapolti L., Project-oriented approach in teaching programming to the first year undergraduate students in Electrical Engineering," 2023 10th International Conference on Modern Power Systems (MPS), Cluj-Napoca, Romania, 2023, 1-4,https://doi.org/10.1109/MPS58874.2023.10187591

11. Lewis JE, Hawkins N., Robinson BS Comparing First-Year Student Programming Confidence Perceptions Between Different Hands-On Projects Paper presented at 2023 ASEE Annual Conference & Exposition, Baltimore, Maryland.https://peer.asee.org/43274
12. Vakaliuk T., Chyzhmotria O., Antoniuk D., Medvediev M. and Didkivska S. The Use of ICT of Educational Purposes in the Independent Work of Students in the Study of the Basics of Programming by Future Software Engineers. In Proceedings of the 2nd Myroslav I. Zhaldak Symposium on Advances in Educational Technology (AET 2021), 644-657https://doi.org/10.5220/0012066700003431
13. Mosobalaje OO, and Orodu OD Introducing Python Coding to Petroleum Engineering Undergraduates: Excerpts from a Teaching Experience." Paper presented at the SPE Nigeria Annual International Conference and Exhibition, Lagos, Nigeria, July 2023,https://doi.org/10.2118/217148-MS
14. Othman J., Mahlan S.B., Kadar R., Shamsuddin M., & Wahab N.A. Factors Affecting Engineering Students in Learning the Programming Subject at UiTM Pulau Pinang: A Study on Educators' Ability. International Journal of Academic Research in Progressive Education and Development, (2023) 12(2), 944–953http://dx.doi.org/10.6007/IJARPED/v12-i2/17359
15. Nguyen A, Ha V. Teaching DevOps and Software Engineering Practices Using an Automated Programming Assessment System. journal [Internet]. 20Jul.2023 Available from:https://journal.tvu.edu.vn/index.php/journal/article/view/2104
16. Pusztai K. Education supported by games even in university? IntechOpen.https://doi.org/10.5772/intechopen.1002208
17. Zakiah N., Khalid D., Hidayah W. A Review of Gamification for Learning Programming Language, International Conference on Business Studies and Education (ICBE 2023)https://www.icbe.my/wp-content/uploads/2023/06/5-1.pdf
18. Zhuang Y., Wang L., Zhang M., Lin S., Hu H. and Tao X. OPTES: A Tool for Behavior-based Student Programming Progress Estimation 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), 122-131,https://doi.org/10.1109/COMPSAC57700.2023.00025.
19. Oosterwyk G., Tsibolane P., Kautondokwa P., Canani A. Beyond the Hype: A Cautionary Tale of ChatGPT in the Programming Classroom. Conference: SACLA2023 https://www.researchgate.net/publication/372495260_Beyond_the_Hype_A_Cautionary_Tale_of_ChatGPT_in_the_Programming_Classroom/link/64ba588fc41fb852dd8d4b11/download
20. Jacques L. Teaching CS-101 at the Dawn of ChatGPT. ACM Inroads (2023) 14(3):40–46https://doi.org/10.1145/3595634
21. Bengtsson D., Kaliff A. Assessment Accuracy of a Large Language Model on Programming Assignments Degree Project in Computer Science and Engineering 2023,https://www.diva-portal.org/smash/get/diva2:1779792/FULLTEXT01.pdf
22. Denny P., Luxton-Reilly A., Leinonen J, Amarouche T., Reeves B., Prather J., Becker B. Promptly: Using Prompt Problems to Teach Learners How to Effectively Utilize AI Code Generators arXiv:2307.16364v1 [cs. HC] 31 Jul 2023,https://arxiv.org/pdf/2307.16364.pdf
23. Zheng Y. ChatGPT for Teaching and Learning: An Experience from Data Science Education. In The 23rd Annual Conference on Information Technology Education (SIGITE '23)https://arxiv.org/pdf/2307.16650.pdf
24. Farah J., Ingram S., Purohit A., Holzer A., and Gillet D. Harnessing Rule-Based Chatbots to Support Teaching Python Programming Best Practices international Conference on Interactive Collaborative Learning (ICL2023), Madrid, Spain, September 26-29 , 2023https://infoscience.epfl.ch/record/303628
25. Dolinsky M. An Approach to Teaching Introductory-Level Computer Programming, Olympiads in Informatics, 2013, 7, 14-22https://ioinformatics.org/journal/INFOL114.pdf
26. Dolinsky M. Teaching Algorithms and Programming First Year University Students on Base of Distance Learning System DL.GSU.BY, WSEAS Transactions on Advances in Engineering Education, 2022a, 19, 52-57https://wseas.com/journals/articles.php?id=5611
27. Dolinsky M. Instrumental System of Distance Learning DL.GSU.BY and Examples of its Application. Global Journal of Computer Science and Technology Interdisciplinary, 2022e, 22(1), 45-53https://globaljournals.org/GJCST_Volume22/6-Instrumental-System-of-Distance-Learning.pdf
28. Dolinsky M. A New Generation Distance Learning System for Programming and Olympiads in Informatics, Olympiads in Informatics, 2017, 11, 29-39 https://ioinformatics.org/journal/v11_2017_29_39.pdf