Buletin Ilmiah Sarjana Teknik Elektro
Vol. 5, No. 4, December 2023, pp. 584-591
ISSN: 2685-9572, DOI: 10.12928/biste.v5i4.9704

Trends in the Development of Basic Computer Education at Universities

Michael Dolinsky
Faculty of Mathematics and Programming Technologies, Francisk Skorina Gomel State University, Gomel, Belarus

ARTICLE INFORMATION

Article History:

Submitted 30 November 2023
Revised 01 January 2024
Accepted 10 January 2024

Keywords:

Literature Review;
Basic Computer Education;
Undergraduate Teaching;
Blended Learning;
DL.GSU.BY

Corresponding Author:

Michael Dolinsky
Faculty of Mathematics and Programming Technologies, Francisk Skorina Gomel State University, Gomel, Belarus.
Email: dolinsky@gsu.by

ABSTRACT

Basic computer education in universities is experiencing huge problems. On the one hand, the amount of knowledge that a university graduate must have is increasing very quickly. On the other hand, the contingent of students varies greatly in terms of the level of training and motivation, and the level of this differentiation is constantly growing. As a result, the complexity of training and the percentage of dropouts increase. Scientists and educators are looking for a solution to these problems in the following areas: revising the knowledge necessary for obtaining at the university in the direction of the reality of their receipt in the allotted time; the use of new information technologies to simplify the learning process and improve its quality; development of the latest teaching methods that take into account the realities. This paper presents a strategic document in the field of computer education at universities - Computing Circulum 2020, as well as an overview of the areas of development of basic computer education, such as learning using artificial intelligence, virtual laboratories, microprocessor kits and robotics, WEB - systems for distance and blended learning, mobile application development, visual programming, gamification, computer architecture & organization, programming languages, learning technologies. In addition, the author gives his experience and vision of teaching basic computer education at universities.

Document Citation:

M. Dolinsky, "Trends in the Development of Basic Computer Education at Universities," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 5, no. 4, pp. 584-591, 2023, DOI: 10.12928/biste.v5i4.9704.

1. INTRODUCTION

Computer programming is one of the core subject of technical universities but the students' performance continues to drop on the following reasons by student opinions: using the conventional static materials, the class should be divided into small groups, educators are concerned with the programming syntax instead of the problem-solving, selecting the wrong pedagogy and programming approaches, the programming language selected is difficult to teach [1].

Challenges and advances in teaching programming and computing thinking are reviwed across the continents: Europe, North America, Oceania, Asia, Latin America and Africa [2], noting that the latter two have a significant number of challenges. The author briefly outlined the goal of teaching programming and computer thinking as the development of five skills:

- abstraction (deciding what information about an entity/object to keep and what to ignore)
- generalization (formulating a solution in generic terms so that it can be applied to different problems)
- decomposition (breaking down a complex problem into smaller parts that are easier to understand and solve)
- algorithms (devising a step-by-step set of operations/actions in order to solve a problem)
- debugging (identifying, removing, and fixing errors).

At [3] there are reviewed modern programming teaching methods, including traditional approaches, Inquiry-Based Learning, Project-based Learning, Collaborative Learning, Online and Interactive Learning, Flipped Classroom, Adaptive Learning.

2. METHODS

2.1. Computing Curricula 2020

On December 31, 2020, the two largest international organizations that bring together computer professionals: the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS) released a joint 205-page document Computing Curricula 2020 (CC 2020) [4], which defines the strategy computer education in seven areas: Computer Engineering, Computer Science, Cybersecurity, Information Systems, Information Technology, Software Engineering, Data Science (under development).

The key difference between CC 2020 and the previous version of the CC 2005 document is the transition from knowledge to competencies. Here is how it is explained in CC 2020:

There is a general agreement in educational circles that career success requires three things.

- Knowledge—"know-what"—a proficiency in core concepts and content and the application of learning to new situations;
- Skills—"know-how"—the ability to carry out tasks with determined results; and
- Dispositions—"know-why"—intellectual, social, or moral tendencies.

Hence, any definition of competency must connect the three dimensions within a context or task represented as: Competency = Knowledge + Skills + Dispositions. CC2020 will help shape the future of computing education. The CC2020 task group followed these four principles in developing CC2020 Report.

1. It must preserve and support the notion of computing, both now and in the future, throughout the world.
2. It must capture future trends and visions from industry, from research, and from across the entire spectrum of society.
3. It must be expansive and support existing, emerging, and future computing programs for its constituents.
4. It must be flexible to achieve global enduring acceptance and be adaptable within multiple educational contexts.

Computing is not just a single area of study, but rather a family of study areas.

- Computer engineering emerging from electrical engineering
- Computer science evolving into a more mature academic discipline
- Information systems expanding as computers became the foundation for organizational processes and work environments
- Information technology emerging as a new discipline that fosters building and maintenance of computing infrastructures
- Software engineering emerging as a discipline based on computer science and computer engineering

The following components can be distinguished as the core of teaching computer knowledge: Computer Architecture and Organization and Programming Languages. Separately, in CC 2020, attention is paid to Computing in Primary and Secondary Education. The paper presents typical examples of work in these areas by various organizations and authors in recent years.

2.2. Learning with Artificial Intelligence

AI programs such as ChatGPT are able to perform tasks in introductory programming courses instead of students [5], while ChatGPT understands not only text descriptions of tasks, but also programs written in various programming languages such as Python, Java, C ++ and etc. Accordingly, teachers are faced with the task of changing the corresponding assignments.

Author [6] presented his vision of teaching an introductory programming course with the help of artificial intelligence (AI). Among other things, he suggests giving tasks to use the AI generating program 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 version of 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 AI-generated solutions.

Using ChatCPT is suggested [7] to grade student-designed programs, noting that historically this has been done either by manual teacher review or automated quizzes. Traditional approaches to teaching programming focus on the practice of writing code [8]. But with the advent of the AI-generator in the code, it is required to change these approaches towards the development of skills in the formation of effective text queries to AI-code generators.

Description of the use of ChatGPT in data science education [9], pointing out personalized learning, concept understanding, coding generation and explanations, educational assessment as the merits of such an approach. Data science includes three essential mandatory courses – a database and data management course, an introductory data analytics course, and a data mining and machine learning course. The author's ChatBot is presented [10], which helps students master the design of programs in Python in accordance with the PEP-8 standard.

2.3. Virtual Laboratories

A new coding platform and method are presented [11]. Students in grades 5-12 use the virtual reality environment to write code to create an interesting world, specifically a dream rollercoaster. As they code, they watch their roller coaster develop in a turn-based 3D environment. Once the roller coaster design is complete, they can deploy it on their smartphone. They can then ride a roller coaster in a virtual environment by placing their phone in a virtual reality headset. The designed and implemented virtual reality platform goes beyond simple visual stimuli and allows students to directly participate and experience their design in a tangible, interactive and operational system. There is presented description [12] of a virtual laboratory based on DSO38Lab virtual instrument experiment box, hardware data acquisition and software supporting graphic display, program debugging, signal generation, time domain and frequency domain analysis, digital filter etc.

2.4. Microprocessor Kits and Robotics

The Arduino platform has become extremely popular at all levels of technical education in the last decade. In addition to cheap hardware, the platform offers a free development environment. There are also additional hardware modules (so-called shields), additional software modules (libraries) and educational materials. An important addition to the Arduino platform is the TinkerCAD simulation and debugging environment, as well as the ability to interact with the MATLAB-Simulink software. A methodology for sharing Arduino, TinkerCAD, MATLAB-Simulink in teaching robotics to students is provided at [13].

Paper [14] describes an Arduino-based project-based learning methodology in which students designed windmill control and water filtration systems. The use of System on a Chip ESP8266 and ESP32 in the educational process is described at [15]. They are compatible with Arduino, have programmable firmware and low cost. During the classes, students master the development of programs using WiFi and BlueTooth.

An example of teaching students how to visually program humanoid robots provided at [16]. Such training is very interesting, especially for beginners. Usually a humanoid robot consists of sensors and actuators such as a body, a head, two arms, two legs. Paper [17] discuss how to teach microcontroller programming beginners more effectively - in a test or graphical environment, comparing the advantages and disadvantages of both approaches.

2.5. WEB systems for distance and blended learning

A WEB platform for teaching Python programming developed by Hausa-speaking countries such as Nigeria, Niger, Sudan, Ghana, and Cameroon is described at [18]. The WEB-platform provides the creation and content of training courses, a forum for communication between students and teachers, and the issuance of certificates of course completion. WEB-platform WEbQuest for learning C programming in the Visual Studio environment is offered at [19].

Paper [20] introduced the WEB platform OPTES: Online Programming Training Estimation System that provides an integrated environment for after-class programming assignments, which is designed to track and estimate the students' learning status continuously over a long period, eg, a whole semester.

The ProgEdu system [21] is developed for automatic grading of students' work in programming, which is close to the development of real software, helping students understand fundamental concepts and gain experience in a modern software development environment. The author's WEB-platform [22] for teaching students how to program logic controllers is presentd. In the near future author anonces development for IoT programming with support for Arduino and ESP32 and STM32 microcontrollers and digital signal processors.

The practice of using the EduCoder platform to teach programming is presented at [23]. Her significant assistance in three aspects of the learning process is noted: organizing classes, designing online experiments, and tracking student progress. Important results are a reduction in the burden on teachers, an increase in the efficiency and quality of education.

2.6. Mobile Application Development

At [24] it is noted that there are a significant number of programming languages and mobile application development tools. And the author describes in detail one of them - Kulika, which he used in teaching students, since it effectively supports all stages of development: analysis, design, development, testing and debugging.

2.7. Visual Programming

There is the opinion that traditional text-based programming languages create difficulties in teaching beginners. As an alternative, the use of visual programming languages is proposed. An example of the use of visual programming languages for the formation of evaluative tests is given [25].

2.8. Gamification

Paper [26] provides an overview of gamification tools for learning programming languages, including incorporation of game mechanics like points, badges, and leaderboards. The feedback and sensation of progress that these game components give students encourages them to keep learning. The author's experience of introducing gamification into university courses "Algorithms and Data Structures" and "Programming methods" is described [27].

2.9. Computer Architecture & Organization

Paper [28] describes the author's course on assembler and low-level C, introduced at MIT since 2022 in the specialties Electrical Engineering and Computer Science, and Computer Science and Engineering.

2.10. Programming Languages

The author's approach [29] to teaching programming and programming languages using C, C++, C# and Python using special engineering tasks for students is described. Paper [30] describes a Python programming course that includes input/output, objects (values, variables, keywords), conditional and repetitive structures, functions, lists, tuples and dictionaries, and application of programs in various subject areas. The author's approach to the study of disciplines related to data science, within which the programming languages R, Python, C ++, SQL are studied at [31]. The paper [32] describes a methodology for teaching the PROLOG programming language.

2.11. Learning technologies

An example of applying the Teaching for Mastery methodology to teaching programming is presented at [33]. Its essence lies in the fact that the student proceeds to study the next topic only after he has mastered the previous one, and each student moves through the educational material at his own speed. The basis of learning is the flipped classroom approach. Theoretical information is contained in prepared short (3-12 minutes) video lectures. To test students' knowledge, an automatic test system is used. Weekly consultations are held for needy students. A methodology for teaching programming using Massive Open Online Courses (MOOC): Sololearn, CodesDope, CodinGeek, Javatpoint, Pskills, Codescracker is described at [34].

3. RESULT AND DISCUSSION

3.1. Author's experience

For several decades, the author has been teaching programming to schoolchildren and students, as well as teaching the basics of digital devices and computer architecture to students of the Faculty of Mathematics and Programming Technologies, specializing in Software, Computer Science and Programming Technologies, and Applied Informatics. To increase the effectiveness of this training, the author developed special methods, and for their full implementation, special software was developed under the guidance of the author.

3.2. Instrumental System of Distance Learning DL.GSU.BY

Since 1999, the distance learning instrumental system DL.GSU.BY [35],[36] has been introduced, operated and continues to develop. It allows you to create training courses, fill them with theory (texts, graphics, videos), add practical tasks (both can be organized in a tree structure). There is a forum for interaction of students among themselves and with teachers. Tests are supported when special tasks are opened for a limited time. A mechanism and interface for interaction with a third-party system that checks solution files sent via the Internet is provided. Thus, in particular, programs in the programming languages Pascal, C, C++, C#, Python, Java, etc. are checked.

3.3. High-Level Chip Computer-Aided Design System

The HLCCAD (High-Level Chip Computer-Aided Design) system allows you to interactively and visually draw functional diagrams of digital devices, simulate their work on input tests (set interactively or in files), analyze simulation results, find and correct errors [37]. The HLCCAD system provides automatic comparison of simulation results with reference results prepared by the author of the job. Thus, informing the student about the correctness / incorrectness of the assignment. In addition, the HLCCAD system is integrated with the DL.GSU.BY system, on the basis of which, in the corresponding training courses, tasks are presented to students through the DL website and through it, students send their schemes for verification. The result of the circuit check by the HLCCAD system is reported to the student and entered in the results table.

3.4. WINTER - Program Debugging Environment for Microprocessors / Microcontrollers

The Winter system allows you to write and debug assembly programs for microcontrollers and microprocessors Intel 8051, Intel 8086, Atmel AT90S2313, Atmel AT90S2323, Motorola M68HC05 and Motorola M68HC08, TMS370 [38]. The Winter system also allows you to configure yourself for other microprocessor/microcontroller models. The Winter system is integrated with the DL.GSU.BY system, on the basis of which, in the corresponding training courses, tasks are presented to students through the DL website and through it, students send their programs for verification. The result of checking the program by the WINTER system is reported to the student and entered in the results table.

3.5. Methodology for Teaching Students the Basics of Digital Devices

The main principles of the methodology for teaching the basics of digital devices [38],[39] and computer architecture [40]:

- practice-oriented approach - the main goal of the course is to teach students how to design circuits and write programs in assembler and C
- automatic verification of solutions (schemes and programs)
- active use of modelling, debugging and visualization tools in order to improve the quality of assimilation of the material
- a significant number (thousands) of learning tasks of varying complexity that support personalized learning.

3.6. Methodology for Teaching Programming to First-Year Students

The main principles of the methodology for teaching programming to first-year students [41]:

- personalized approach to learning (student choice):
  - programming languages Pascal, C, C++, C#, Java, Python;
  - the level of complexity of tasks in the current training
- practice-oriented approach - the main goal of the course is to teach the student to write and debug programs in their chosen programming language
- automated objective control and assessment of knowledge
- the main functions of the teacher
  - to teach the student to learn
  - to provide individual assistance to students
- the presence of all the necessary material in the training course at various levels of abstraction

3.7. Methodology for Teaching Programming to High School Students (Grade 9-11)

The author absolutely agrees with Computing Cirricula 2020 [4] that in order to get high-quality students at a university, one must work with schoolchildren and the author has been doing this all his life. The main principles of the methodology for teaching programming to high school students [42]:

- preparation for victories in Olympiads in Informatics: international, national, regional. The level of achievements depends not only and not so much on abilities, but on the invested work and time spent
- sequential training from simple to complex with permanent verification of the effectiveness of training - weekly in training Olympiads and monthly in rounds on Codeforces.com
- a significant amount of individual independent work of the student.

3.8. Methodology for Teaching Programming in secondary school (Grade 5-8)

In order for older students to be able to successfully compete in international and national olympiads, they must start classes at least in secondary school. Basic principles for teaching programming to secindary school students [43]:

- personalized adaptive learning
- regular participation in training and official Olympiads
- development of independence and creativity

3.9. Methodology for Teaching Programming in Primary School (Grade 1-4)

In contrast to the dominant approach of block visual programming with beginners (not only in primary and secondary school, but even in high school), the author adheres to the opposite point of view that learning can and should be started with text programming, and all emerging problems can be solved with the help of an appropriate methodology and software [44]-[46]. Moreover, under the guidance of the author, special programming competitions for elementary school students have been developed and are held annually [47]. To equalize the levels of preparation of first-graders, as well as to be able to work with preschoolers, a special course “Learning to Think” was developed and is being used [48].

4. CONCLUSIONS

This paper provides an overview of the areas of development of basic computer education, such as: training with the help of artificial intelligence, virtual laboratories, microprocessor kits and robotics, WEB systems for distance and blended learning, mobile application development, visual programming, gamification, computer architecture & organization , programming languages, learning technologies. In addition, the author gives his experience and vision of teaching basic computer education at universities, considering such issues as specially designed software and his own methods of teaching the basics of digital technology and programming

REFERENCES

[1] J. Othman S. B. Mahlan R. Kadar M. Shamsuddin and N. A. Wahab, “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*, vol. 12, no. 2, pp. 944–953, 2023, https://doi.org/10.6007/IJARPED/v12-i2/17359.

[2] H. Belmar, “Review on the Teaching of Programming and Computational Thinking in the World,” *Frontiers in Computer Science*, vol. 4, p. 997222, 2022, https://doi.org/10.3389/fcomp.2022.997222.

[3] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, “Metaheuristic algorithms: A comprehensive review,” *Computational intelligence for multimedia big data on the cloud with engineering applications*, pp. 185-231, 2018, https://doi.org/10.1016/B978-0-12-813314-9.00010-4.

[4] F. Almeida and J. Simoes, “The role of serious games, gamification and industry 4.0 tools in the education 4.0 paradigm,” *Contemporary Educational Technology*, vol. 10, no. 2, pp. 120-136, 2019, https://doi.org/10.30935/cet.554469.

[5] A. Azaria, R. Azoulay, and S. Reches, “ChatGPT is a Remarkable Tool--For Experts,” *arXiv preprint arXiv:2306.03102*, 2023, https://doi.org/10.48550/arXiv.2306.03102.

[6] L. Jacques, “Teaching CS-101 at the Dawn of ChatGPT,” *ACM Inroads*, vol. 14, no. 2, pp. 40-46, 2023, https://doi.org/10.1145/3595634.

[7] M. Pankiewicz and R. S. Baker, “Large Language Models (GPT) for automating feedback on programming assignments,” *arXiv preprint arXiv:2307.00150*, 2023, https://doi.org/10.48550/arXiv.2307.00150.

[8] P. Denny, J. Leinonen, J. Prather, A. Luxton-Reilly, T. Amarouche, B. A. Becker, and B. N. Reeves, “Promptly: Using prompt problems to teach learners how to effectively utilize ai code generators,” *arXiv preprint arXiv:2307.16364*, 2023, https://doi.org/10.48550/arXiv.2307.16364.

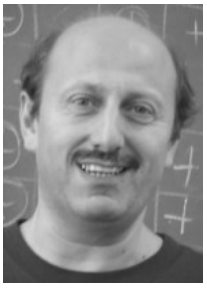
[9] Y. Zheng, “ChatGPT for Teaching and Learning: An Experience from Data Science Education,” In *Proceedings of the 24th Annual Conference on Information Technology Education*, pp. 66-72, 2023, https://doi.org/10.1145/3585059.3611431.

[10] L. Brehm, H. Guenzel, O. Hinz, A. Humpe and H. Martius, "Collaborative Learning with COZMO to Teach Programming in SCRATCH and Python," *2019 IEEE Global Engineering Education Conference (EDUCON)*, pp. 448-452, 2019, https://doi.org/10.1109/EDUCON.2019.8725037.

[11] M. Zargham and J. Cheng, "A Virtual Reality Programming Platform for Grades 5-12," In *EdMedia+ Innovate Learning*, pp. 1067-1073, 2023, https://www.learntechlib.org/primary/p/222616/.
[12] X. Xu, "Teaching Reform of Virtual Instrument Technology and Application Course Based on OBE Concept," *Open Access Library Journal*, vol. 10, no. 8, pp. 1-9, 2023, https://doi.org/10.4236/oalib.1110491.
[13] R. Čermák, V. Mašek, and D. Fait, "TEACHING MECHATRONICS, ROBOTICS AND AUTOMATION WITH ARDUINO," In *EDULEARN23 Proceedings*, pp. 6497-6502, 2023, https://doi.org/10.21125/edulearn.2023.1720.
[14] F. Martin, B. Stamper, and C. Flowers, "Examining Student Perception of Readiness for Online Learning: Importance and Confidence," *Online Learning*, vol. 24, no. 2, pp. 38-58, 2020, https://doi.org/10.24059/olj.v24i2.2053.
[15] N. C. Cruz, J. S. Estrada, M. Lupión, E. M. Garzón, and P. M. Ortigosa, "LOW-COST CHIP PROGRAMMING FOR TEACHING AND LEARNING DIGITAL SKILLS," In *EDULEARN23 Proceedings*, pp. 2662-2671, 2023, https://doi.org/10.21125/edulearn.2023.0763.
[16] M. Zimmermann, "Teaching Visual Programming: Humanoid Robot Programming as a Case Study," In *EDULEARN23 Proceedings*, pp. 6143-6149, https://doi.org/10.21125/edulearn.2023.1598.
[17] F. Vrbančič and S. Kocijančič, "Strategy for learning microcontroller programming—a graphical or a textual start?," *Education and Information Technologies*, pp. 1-23, 2023, https://doi.org/10.1007/s10639-023-12024-9.
[18] M. Mamman, R. S. Rdadi, M. A. Sadiq, "Computerized E-learning platform for teaching Python programming using Hausa language," *World Journal of Advanced Engineering Technology and Sciences*, vol. 9, no. 2, pp. 116–123, 2023, https://doi.org/10.30574/wjaets.2023.9.2.0214.
[19] I. Aldalur and X. Sagarna, "Improving Programming Learning in Engineering Students Through Discovery Learning," in *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 18, no. 3, pp. 239-249, 2023, https://doi.org/10.1109/RITA.2023.3301409.
[20] Y. Zhuang, L. Wang, M. Zhang, S. Lin, H. Hu and X. Tao, "OPTES: A Tool for Behavior-based Student Programming Progress Estimation," *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 122-131, 2023, https://doi.org/10.1109/COMPSAC57700.2023.00025.
[21] K. Kuusinen and S. Albertsen, "Industry-Academy Collaboration in Teaching DevOps and Continuous Delivery to Software Engineering Students: Towards Improved Industrial Relevance in Higher Education," *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, pp. 23-27, 2019, https://doi.org/10.1109/ICSE-SEET.2019.00011.
[22] I. Fonseca, N. C. Martins and F. Lopes, "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)*, pp. 1-6, 2022, https://doi.org/10.1109/EAEEIE54893.2022.9820594.
[23] S. Zhang, J. Yang, and X. Sang, "Exploring the Applications of EduCoder Platform in Blended Teaching for Computer Major," *Journal of Education and Educational Research*, vol. 4, no. 2, pp. 100-103, 2023, https://doi.org/10.54097/jeer.v4i2.10819.
[24] İ. H. Özdemır, T. Kişla and S. S. Tito, "Mobile Programming with Kuika: A Course Design," *2023 32nd Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE)*, pp. 1-6, 2023, https://doi.org/10.23919/EAEEIE55804.2023.10181754.
[25] Y. P. Cheng, S. C. Cheng, M. Yang, J. M. Lin, and Y. M. Huang, "Applying Computational Thinking and Formative Assessment to Enhance the Learning Performance of Students in Virtual Programming Language," In *International Conference on Innovative Technologies and Learning*, pp. 130-139, 2023, https://doi.org/10.1007/978-3-031-40113-8_13.
[26] M. Shahid, A. Wajid, K. U. Haq, I. Saleem and A. H. Shujja, "A Review of Gamification for Learning Programming Fundamental," *2019 International Conference on Innovative Computing (ICIC)*, pp. 1-8, 2019, https://doi.org/10.1109/ICIC48496.2019.8966685.
[27] N. Campos, M. Nogal, C. Caliz, and A. A. Juan, "Simulation-based education involving online and on-campus models in different European universities," *International Journal of Educational Technology in Higher Education*, vol. 17, pp. 1-15, 2020, https://doi.org/10.1186/s41239-020-0181-y.
[28] F. J. Gallego-Durán, R. Satorre-Cuerda, P. Compañ-Rosique, C. J. Villagrá-Arnedo, R. Molina-Carmona, and F. Llorens-Largo, "A low-level approach to improve programming learning," *Universal Access in the Information Society*, *20*, 479-493, 2021, https://doi.org/10.1007/s10209-020-00775-y.
[29] L. Grindei, C. Constantinescu, A. Bojita, R. Holonec and L. Rapolti, "Project-oriented approach in teaching programming to the first year undergraduate students in Electrical Engineering," *2023 10th International Conference on Modern Power Systems (MPS)*, pp. 1-4, 2023, https://doi.org/10.1109/MPS58874.2023.10187591.
[30] O. O. Mosobalaje and O. D. Orodu, "Introducing Python Coding to Petroleum Engineering Undergraduates: Excerpts from a Teaching Experience," In *SPE Nigeria Annual International Conference and Exhibition*, p. D032S028R003, 2023, https://doi.org/10.2118/217148-MS.
[31] T. Donoghue, B. Voytek, and S. E. Ellis, "Teaching creative and practical data science at scale," *Journal of Statistics and Data Science Education*, vol. 29, no. sup1, pp. S27-S39, 2021, https://doi.org/10.1080/10691898.2020.1860725.
[32] H. Tambunan, "The Effectiveness of the Problem Solving Strategy and the Scientific Approach to Students' Mathematical Capabilities in High Order Thinking Skills," *International electronic journal of mathematics education*, vol. 14, no. 2, pp. 293-302, 2019, https://doi.org/10.29333/iejme/5715.
[33] M. Zappatore, "Incorporating Collaborative and Active Learning Strategies in the Design and Deployment of a Master Course on Computer-Assisted Scientific Translation," *Technology, Knowledge and Learning*, pp. 1-56, 2023, https://doi.org/10.1007/s10758-023-09679-1.

[34] I. S. Zinovieva *et al*., "The use of online coding platforms as additional distance tools in programming education," In *Journal of physics: Conference series*, vol. 1840, no. 1, p. 012029, 2021, https://doi.org/10.1088/1742-6596/1840/1/012029.
[35] M. Dolinsky, "A new generation distance learning system for programming and Olympiads in Informatics," *Olympiads in Informatics*, vol. 11, pp. 29-40, 2017, https://doi.org/10.15388/ioi.2017.03.
[36] 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), 45-53, 2022, https://globaljournals.org/GJCST_Volume22/6-Instrumental-System-of-Distance-Learning.pdf.
[37] M. S. Dolinsky, "Tool HLCCAD for Blended Learning the Fundamentals of Digital Electronics. *International Journal of Circuits and Electronics*, vol. 7, 2022, https://www.iaras.org/iaras/home/computer-science-communications/caijce/tool-hlccad-for-blended-learning-the-fundamentals-of-digital-electronics.
[38] M. Dolinsky, "Experience of Blended Learning the Fundamentals of Digital Electronics for First/Second Year University students On Base of Distance Learning System DL.GSU.BY," *International Journal of Education and Learning Systems*, vol. 7, 2022, http://elib.gsu.by/jspui/handle/123456789/43782.
[39] M. Dolinsky, "Teaching Algorithms and Programming First Year University Students on Base of Distance Learning System DL.GSU.BY," *WSEAS Transactions on Advances in Engineering Education*, vol. 19, pp. 52-57, 2022, https://doi.org/10.37394/232010.2022.19.6.
[40] M. Dolinsky, "Experience of Blended Learning in the Subject - Architecture of Computers," *Journal of Information Technology and Digital World*, vol. 4, no. 3, pp. 167-182, 2022, https://doi.org/10.36548/jitdw.2022.3.004.
[41] M. Dolinsky, "An Approach to Teaching Introductory-Level Computer Programming," *Olympiads in Informatics,* vol. 7, pp. 14-22, 2013, https://ioinformatics.org/journal/INFOL114.pdf.
[42] M. Dolinsky, "Gomel Training School for Olympiads in Informatics," *Olympiads in Informatics,* vol. 10, pp. 237-247, 2016, https://doi.org/10.15388/ioi.2016.16.
[43] M. Dolinsky, "Secondary School Programming Olympiads in Gomel Region (Belarus)," *Olympiads in Informatics,* vol. 17, pp. 131-142, 2023, https://doi.org/10.15388/ioi.2023.10.
[44] N. Dolinsky and M. Dolinskaya, "How to Start Teaching Programming at Primary School," *Olympiads in Informatics,* vol. 12, pp. 13-24, 2018, https://doi.org/10.15388/ioi.2018.02.
[45] M. Dolinsky and M. Dolinskaya, "Training in Writing the Simplest Programs from Early Ages," *Olympiads in Informatics*, vol. 13, pp. 21-30, 2019, https://doi.org/10.15388/ioi.2019.02.
[46] M. Dolinsky and M. Dolinskaya, "The Technology of Differentiated Instruction in Text Programming in Elementary School Based on the Website DL.GSU.BY," *Olympiads in Informatics,* vol. 14, pp. 37-46, 2020, https://doi.org/10.15388//ioi.2020.03.
[47] M. Dolinsky, "Primary School Programming Olympiads in Gomel Region (Belarus)," *Olympiads in Informatics,* vol. 16, pp. 107-123, 2022, https://doi.org/10.15388/ioi.2022.09.
[48] M. Dolinsky, "Technology for the Development of Thinking of Preschool Children and Primary School Pupils," *Olympiads in Informatics*, vol. 8, pp. 63-68, 2014, https://ioi.te.lv/oi/pdf/v8_2014_63_68.pdf.

AUTHOR BIOGRAPHY



Michael Dolinsky is a lecturer in Gomel State University "Fr. Skoryna" from 1993. Since 1999 he is a leading developer of the educational site of the University (dl.gsu.by). Since 1997 he is heading preparation of the scholars in Gomel to participate in programming contests and Olympiad in informatics. He was a deputy leader of the team of Belarus for IOI'2006, IOI'2007, IOI'2008 and IOI'2009. His PhD is devoted to the tools for digital system design. His current research is in teaching Computer Science and Mathematics from early age.