AMERICAN JOURNAL OF EDUCATION AND LEARNING
ISSN: 2996-5128 (online) | ResearchBib (IF) = 10.91 IMPACT FACTOR
Volume-4| Issue-2| 2026 Published: |28-02-2026|

FORMS AND STAGES OF DEVELOPING PROGRAMMING COMPETENCIES OF HIGHER EDUCATION STUDENTS BASED ON A SYSTEMATIC-ALGORITHMIC APPROACH

https://doi.org/10.5281/zenodo.18500637

Khazratov Fazliddin Khikmatovich
Bukhara State University, Doctor of Philosophy (PhD), Associate Professor

Abstract

This article presents the forms and stages of developing programming competencies of higher education students based on a systematic-algorithmic approach. The study analyzes the process of forming students' programming skills based on modern educational technologies and pedagogical approaches. The advantages of the systematic-algorithmic approach and ways of its practical application are demonstrated. The research results were confirmed through experimental work.

Keywords

programming competence, systematic-algorithmic approach, students, development forms, stage-based education, higher education, curriculum.

INTRODUCTION

Currently, the rapid development of information technologies and the expansion of digitalization processes place new demands on the higher education system, particularly on programming education. The Decree of the President of the Republic of Uzbekistan No. PF-5992 dated April 29, 2020, "On Additional Measures for the Development of Digital Economy and E-Government in the Information and Communication Sphere" and Decree No. PF-5953 dated February 17, 2020, "On the Improved Development Strategy of the Republic of Uzbekistan for 2020-2030" have further increased the demand for specialists in the field of programming.

Among foreign researchers on the development of programming competencies, J. Wing (2006), M. Guzdial (2015), A. Robins and J. Rountree (2003) focused on developing methods for teaching algorithmic thinking and programming. Among local scientists, S.S. Gulomov (2020), N.A. Sayidakhmedov (2019), A.A. Abdukodirov (2018) and others studied the application of modern technologies in education.

Research objective: to develop an effective system of pedagogical forms and stages for developing programming competencies of higher education students

AMERICAN JOURNAL OF EDUCATION AND LEARNING
ISSN: 2996-5128 (online) | ResearchBib (IF) = 10.91 IMPACT FACTOR
Volume-4| Issue-2| 2026 Published: |28-02-2026|

based on a systematic-algorithmic approach and to determine its practical effectiveness.

METHODOLOGY

The following methods were used in the research:

- Theoretical methods: analysis, generalization, and modeling of pedagogical, psychological, and technical literature;
- Empirical methods: pedagogical observation, questionnaires, testing, expert evaluation;
- Experimental methods: pedagogical experiment, ascertaining and formative experiments;
- Statistical methods: mathematical-statistical analysis, Student's t-test.

The research was conducted at Tashkent University of Information Technologies and Jizzakh Branch of the National University of Uzbekistan during the 2022-2024 academic years. 124 students participated in the experimental group and 118 students in the control group. The students were 2nd-3rd year learners studying "Fundamentals of Programming," "Object-Oriented Programming," and "Data Structures and Algorithms" courses.

RESULTS AND DISCUSSION

Theoretical foundations of the systematic-algorithmic approach

The systematic-algorithmic approach involves viewing the educational process as a holistic system and organizing it in a logical sequence. Based on this approach, the following principles were established for developing programming competencies:

- Principle of systematicity – viewing the educational process as a set of interconnected elements;
- Principle of stage-based learning – organizing educational material on the basis of transition from simple to complex levels;
- Principle of algorithmization – solving each problem in a clear sequence;
- Principle of activity – ensuring independent and active student participation.

Model of development stages

During the research, a 4-stage model was developed for developing students' programming competencies (Figure 1):

AMERICAN JOURNAL OF EDUCATION AND LEARNING
ISSN: 2996-5128 (online) | ResearchBib (IF) = 10.91 IMPACT FACTOR
Volume-4| Issue-2| 2026 Published: |28-02-2026|

Figure 1. Stages of developing programming competencies

Each stage has its own learning objectives, content, and assessment criteria. The sequence of stages ensures the step-by-step development of students' knowledge and skills.

Educational forms and methods

The following educational forms were applied to implement the systematic-algorithmic approach (Table 1):

Table 1

Educational forms and methods by stages

Stage	Educational Forms	Methods	Tools
Stage I	Lecture, practical session, conversation	Visualization, problem-based presentation, game technologies	Multimedia, video lessons
Stage II	Laboratory work, seminar, small group work	Solving algorithmic problems, flowcharts, pseudocode writing	Scratch, Flowgorithm, PascalABC
Stage III	Practical class, independent work, distance learning	Programming languages, debugging, refactoring, code review	Python, C++, Visual Studio, GitHub
Stage IV	Project work, hackathon,	Team projects, design	Django, React,

AMERICAN JOURNAL OF EDUCATION AND LEARNING
ISSN: 2996-5128 (online) | ResearchBib (IF) = 10.91 IMPACT FACTOR
Volume-4| Issue-2| 2026 Published: |28-02-2026|

	internship	patterns, agile methodology	Docker, Git

Experimental results

The pedagogical experiment was conducted in two stages during the 2022-2024 academic years. During the ascertaining experiment stage, students' initial competency levels were determined. During the formative experiment, the developed methodology was applied in the experimental group, while traditional methods were used in the control group.

Students' programming competencies were assessed according to 4 criteria:

- Theoretical knowledge (programming concepts, syntax, data structures);
- Algorithmic thinking (problem analysis, algorithm design);
- Practical skills (code writing, debugging, optimization);
- Creative ability (independent project creation, problem-solving).

Each criterion was assessed on a 100-point scale. Competency levels: low (0-64 points), medium (65-84 points), high (85-100 points). The experimental results are presented in Table 2.

Table 2

Experimental results (in percentages)

Group / Stage	Theoretical Knowledge	Algorithmic Thinking	Practical Skills	Creative Ability
EG (initial)	62.4	58.7	55.3	51.2
EG (final)	**86.3**	**83.9**	**81.7**	**78.5**
CG (initial)	61.8	59.1	56.2	52.4
CG (final)	71.5	68.9	66.3	63.8

Note: EG – Experimental Group, CG – Control Group

The results show significant improvement in all criteria in the experimental group. Theoretical knowledge increased by 23.9%, algorithmic thinking by 25.2%, practical skills by 26.4%, and creative ability by 27.3%. The control group showed much lower growth: 9.7%, 9.8%, 10.1%, and 11.4% respectively.

Statistical analysis using Student's t-test was conducted. The differences across all criteria were found to be statistically significant ($p<0.05$). This confirms the effectiveness of the systematic-algorithmic approach.

CONCLUSION

Based on the research results, the following conclusions were formed:

1. The systematic-algorithmic approach is an effective pedagogical foundation for developing students' programming competencies. This approach allows viewing all components of the educational process (goals, content, forms, methods, tools, results) as a holistic system and organizing them in a logical sequence.

2. The developed 4-stage model (motivational-preparatory, cognitive-algorithmic, practical-technological, creative-project based) ensures the step-by-step development of students' competencies. Each stage is distinguished by its learning objectives, content, forms, methods, and tools.

3. The results of the pedagogical experiment demonstrated the high effectiveness of the systematic-algorithmic approach. The competency level of students in the experimental group increased 2.5 times more than in the control group. This difference was found to be statistically significant ($p<0.05$) across all assessment criteria (theoretical knowledge, algorithmic thinking, practical skills, creative ability).

4. The research results can be applied to improve the teaching process of programming courses in higher education institutions and to develop students' professional competencies. The developed methodology can also be effectively applied in teaching other technical disciplines.

REFERENCES:

1. Decree of the President of the Republic of Uzbekistan No. PF-5992 dated April 29, 2020 "On Additional Measures for the Development of Digital Economy and E-Government in the Information and Communication Sphere."

2. Wing, J.M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

3. Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. Morgan & Claypool Publishers, 165 p.

4. Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. Computer Science Education, 13(2), 137-172.

5. Gulomov, S.S., & Sayidakhmedov, N.A. (2020). Improving the effectiveness of education based on information and communication technologies. Tashkent: Fan, 256 p.

6. Abdukodirov, A.A. (2018). Technologies for forming students' programming competencies. Tashkent: O'qituvchi, 184 p.

7. Papert, S. (2020). Mindstorms: Children, computers, and powerful ideas. Basic Books, 252 p.

8. Resnick, M. (2017). Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play. MIT Press, 200 p.

9. Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. In Emerging research, practice, and policy on computational thinking (pp. 205-220). Springer.

10. Begimkulov, U.Sh. (2019). Pedagogy of applying information technologies in higher education. Tashkent: Iqtisod-moliya, 312 p.