

# Methods Of Teaching Programming at The University

Markhamat Y. Khaydarova

TUIT Associate Professor, Tashkent University Of Information Techmologies Named After Muhammad Al-Khwarizmi, Uzbekistan

## Annotation:

This article explores five effective methods for teaching programming at the university level, aiming to address the diverse needs of students and enhance their learning experience. The five methods discussed include interactive learning, project-based education, peer learning and collaboration, the use of tools and simulators, and flipped classroom techniques. Each method is analyzed in terms of its benefits and practical implementation in the classroom environment. Interactive learning emphasizes active student participation through coding exercises and immediate feedback. Project-based education focuses on real-world application by assigning students hands-on projects that develop both technical and soft skills.

**Key words:** Methods, interactive learning, project-based learning, Peer learning, Tools and simulators, Flipped classroom

- Interactive Learning

Interactive learning is an approach that encourages active participation from students during the learning process. Unlike traditional lecture-based methods where students passively receive information, interactive learning involves hands-on exercises, immediate feedback, and collaboration.

In programming education, interactive learning can be achieved by incorporating coding exercises during lectures or classes. For example, using platforms like Codewars or LeetCode, instructors can provide students with immediate challenges related to the topics being discussed. This allows students to practice coding as they learn, solidifying concepts in real time. Moreover, tools such as Jupyter Notebooks allow for interactive demonstrations where students can experiment with code, tweak variables, and immediately see the effects.

Interactive learning promotes engagement, critical thinking, and retention of knowledge, making it a highly effective method for teaching programming.

- Project-Based Education

Project-based education (PBE) is a teaching method in which students gain knowledge and skills by working on real-life projects. In the context of programming, this approach encourages students to apply what they have learned to solve real-world problems, fostering a deeper understanding of how programming concepts are used in practical scenarios.

For example, students may be tasked with developing a simple web application, building a small database, or writing software that automates a specific task. The instructor's role is to guide the students through the project, providing support and clarification as needed while allowing the students to work independently or in groups.

Through project-based education, students develop not only technical skills but also soft skills such as teamwork, communication, and time management. Moreover, working on projects that mimic real-world applications prepares students for careers in programming by simulating the environments they will encounter in the workforce.

- Peer Learning and Collaboration

Peer learning is another effective method that has proven useful in teaching programming. This approach encourages students to collaborate, discuss ideas, and help one another in understanding complex topics.

In programming, pair programming is a popular peer-learning technique where two students work together on a coding problem. One student writes the code while the other reviews it, providing immediate feedback. This process enhances understanding, helps to identify errors early, and encourages collaborative problem-solving.

Additionally, study groups or online coding communities can serve as platforms for students to share their knowledge, discuss challenges, and learn from each other's experiences. This method fosters a supportive learning environment, and students often find it easier to approach peers with questions or uncertainties than instructors.

#### 1. Project-Based Learning (PBL):

PBL is a student-centered approach where learners actively engage in real-world projects. This method encourages:

- **Deep Learning:** Students delve into the complexities of a project, fostering a deeper understanding of programming concepts.
- **Collaboration:** Working in teams develops essential communication, teamwork, and conflict resolution skills.
- **Problem-Solving:** Real-world scenarios demand creative solutions, honing students' problem-solving abilities.
- **Motivation:** Students find projects more engaging and relevant than traditional lectures, promoting intrinsic motivation.

Example: In a web development course, students could be tasked with creating a functional e-commerce website. They would learn various programming languages, web design principles, and database management, while also facing real-world challenges like user experience and security.

#### 2. Active Learning Techniques:

Active learning strategies move students beyond passive listening and note-taking, encouraging active participation and engagement:

- **Pair Programming:** Students work in pairs, taking turns writing code and providing feedback, promoting collaboration and learning from each other.
- **Group Discussions:** Encouraging discussions about programming concepts and challenges deepens understanding and encourages critical thinking.
- **Interactive Coding Exercises:** Online platforms like Codewars and HackerRank provide interactive coding challenges that allow students to test their skills and receive immediate feedback.
- **Gamification:** Integrating game mechanics into learning materials can make coding more engaging and fun, promoting motivation and competition.

#### 3. Flipped Classroom:

This method reverses the traditional lecture-based model:

- **Pre-Lecture Material:** Students engage with pre-recorded lectures, readings, or online tutorials before coming to class.
- **In-Class Application:** Class time is dedicated to applying knowledge through interactive activities, group work, and problem-solving sessions.

Benefits:

- **Personalized Learning:** Students can learn at their own pace, re-watching lectures or seeking additional resources as needed.
- **Time Efficiency:** Allows instructors to focus on individual student needs and provide more personalized support during class time.

#### 4. Problem-Solving and Debugging:

Beyond simply writing code, teaching students to debug and troubleshoot is crucial for their success as programmers.

- **Collaborative Debugging:** Students work together to identify and resolve errors in code, promoting problem-solving skills and collaborative learning.
- **Code Review:** Peer review and instructor feedback provide valuable insight into coding practices and help students identify areas for improvement.

#### 5. Integrating Emerging Technologies:

The world of programming is constantly evolving, and incorporating emerging technologies into the curriculum is essential.

- Artificial Intelligence (AI) and Machine Learning: Introducing students to basic AI concepts and machine learning algorithms can open doors to exciting career opportunities.
- Data Science and Analytics: Teaching students how to analyze and visualize data can be a valuable skill in many industries.
- Cloud Computing: Providing students with experience with cloud platforms and services prepares them for the modern workplace.

### **Conclusion:**

Teaching programming effectively at the university level requires a departure from traditional methods. Incorporating project-based learning, active learning techniques, flipped classrooms, problem-solving skills, and emerging technologies allows students to develop the knowledge, skills, and mindset necessary to thrive in the ever-evolving world of programming. By embracing these modern methods, universities can equip their graduates with the tools they need to become successful and innovative programmers.

### **References**

1. Kelleher C., Pausch R. Dasturlash uchun to'liq dasturlashni kamaytirish: Ajam dasturchilar uchun dasturlash muhiti va tillarining taksonomiyasi. ACM Computing Surveys, 37(2), 2008.
2. Pilar ST., Rubén FF. Universitet dasturlash kurslarida jamoada ishlash ko'nikmalarini o'rganish // Kompyuterlar va ta'lim , 2009.
3. Strijbos J.-V. Rollarning kompyuter yordamida hamkorlikda o'rganishga ta'siri. Doktorlik dissertatsiyasi. Heerlen, Niderlandiya: Ochiq universitet Niderlandiya, 2014.
4. Daly T. Maksimallashtirish uchun minimallashtirish: Alice yordamida kirish dasturlashni o'rgatishning dastlabki urinishi // Kollejlarda kompyuter fanlari jurnali, 26 (5), bet. 2011 yil 23–30 may. TSPU byulleteni. 2017.
5. Tillmann N., Halleux JD, Xie T. Pex4Fun // Ijtimoiy o'yinlar orqali kompyuter fanlarini o'qitish va o'rganish, 2012 yilda IEEE 25-konferentsiyasi Software Engineering Education and Training (CSEE&T), Nankin, 2012.