ORIGINAL ARTICLE

INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE

ISSN: 2692-5206, Impact Factor: 12,23
American Academic publishers, volume 05, issue 09,2025

Journal: https://www.academicpublishers.org/journals/index.php/ijai

THE METHODOLOGY OF TEACHING FUNCTIONS IN PYTHON PROGRAMMING LANGUAGE

Nazarova Munojat Raxmonberdi kizi

Muhammad al-Xorazmiy nomidagi ixtisoslashtirilgan maktabi Guliston filiali

Abstract: This article explores the methodology of teaching functions in the Python programming language, emphasizing their didactic and practical significance in programming education. Functions are one of the fundamental constructs of programming; they ensure code reusability, modularity, and clarity while fostering abstract thinking skills among learners. The study discusses pedagogical approaches for introducing functions, the effectiveness of using them in practical lessons, and their role in developing students' problem-solving abilities. By analyzing modern educational practices, this paper highlights strategies for enhancing the teaching of Python functions in both secondary and higher education.

Keywords: Python, programming education, functions, teaching methodology, modular programming, algorithm, competencies.

Introduction

Programming education has become a crucial component of modern curricula at schools, colleges, and universities. Among the numerous programming languages, Python has emerged as one of the most widely used due to its simple syntax, readability, and versatility. Its growing popularity in academia, research, and industry makes it an ideal language for teaching fundamental programming concepts.

One of the core elements in Python programming education is the concept of functions. Functions allow programmers to write reusable code, divide complex problems into manageable subproblems, and improve the readability and maintainability of programs. From a pedagogical perspective, teaching functions helps learners develop algorithmic thinking, abstraction skills, and problem decomposition abilities, which are essential for building programming competence.

However, many students face difficulties when first introduced to functions, such as understanding parameter passing, scope of variables, and the difference between built-in and user-defined functions. Therefore, effective teaching methodology must focus on step-by-step instruction, practical examples, and interactive learning environments. This article seeks to examine the methods of teaching Python functions in a structured, student-centered manner and provide recommendations for educators.

Methods

The study is based on a mixed-method approach, combining a review of existing literature on programming pedagogy with classroom-based observations. A total of 60 students from secondary and university-level computer science programs were observed during Python programming courses between 2022 and 2024. Data collection involved:

ORIGINAL ARTICLE
INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE
ISSN: 2692-5206, Impact Factor: 12,23
American Academic publishers, volume 05, issue 09,2025
Journal: https://www.academicpublishers.org/journals/index.php/ijai

- Classroom observations during lectures and lab sessions.
- Student performance analysis in problem-solving tasks involving Python functions.
- Surveys and feedback forms to assess students' attitudes and challenges in learning functions.
- Comparative analysis of different teaching methods, including lecture-based teaching, problem-based learning (PBL), and project-based learning.

Collected data were analyzed thematically to determine the most effective strategies for teaching functions in Python programming.

Results

The study revealed several key findings:

1. **Step-by-step introduction** of functions, starting from built-in functions to user-defined ones, improves comprehension.
2. **Practical tasks and mini-projects** enhance students' ability to apply functions in real-world problem solving.
3. **Visualization tools** (e.g., code tracing software) help students understand parameter passing and variable scope.
4. **Collaborative learning** methods, such as pair programming, encourage deeper engagement and peer-to-peer explanation.
5. Students who engaged in **project-based learning** demonstrated higher retention of knowledge and stronger problem-solving competencies compared to those taught only with lectures.

Discussion

The results highlight that teaching functions in Python requires a balance between theoretical explanation and hands-on practice. Abstract concepts such as recursion, higher-order functions, and lambda expressions must be introduced gradually, with sufficient examples and exercises. Project-based learning emerged as the most effective approach because it provides a context for students to see the practical application of functions in real programming tasks.

Furthermore, the integration of digital learning platforms, such as interactive coding environments (Jupyter Notebooks, Replit, or online compilers), creates opportunities for immediate feedback and active experimentation. These tools help students overcome common difficulties in understanding parameter passing and debugging errors.

Pedagogically, teachers should emphasize **problem decomposition** as the primary rationale for using functions. By guiding students to break down complex algorithms into smaller, reusable modules, educators can reinforce both programming efficiency and logical reasoning skills.

Conclusion

Teaching functions in the Python programming language is not only about introducing a programming construct but also about nurturing critical thinking, abstraction, and modular

ISSN: 2692-5206, Impact Factor: 12,23
American Academic publishers, volume 05, issue 09,2025
Journal: https://www.academicpublishers.org/journals/index.php/ijai

problem-solving skills among students. The research has shown that functions serve as a pedagogical bridge between basic programming concepts and more advanced topics such as recursion, object-oriented programming, and software design. Therefore, developing effective methodologies for teaching functions is crucial in building a strong foundation for learners in computer science.

The findings emphasize that **step-by-step instruction**, supported by practical exercises, plays an essential role in ensuring students understand the logic and application of functions. The gradual progression from built-in to user-defined functions, followed by more complex structures like higher-order functions, allows learners to build confidence and mastery. In addition, **project-based learning** proved to be the most effective approach because it contextualizes the use of functions in real-life problem-solving scenarios, thereby increasing motivation and knowledge retention.

Another important aspect is the use of **visualization tools and interactive coding environments**. These digital platforms provide immediate feedback, enabling students to detect and correct errors while reinforcing their conceptual understanding. Pair programming and collaborative learning methods further enhance student engagement, as learners actively exchange knowledge and support each other in overcoming difficulties.

At a broader level, the methodology of teaching functions in Python must adapt to the rapid digitalization of education. The inclusion of online compilers, gamified learning platforms, and simulation-based environments can make the process more dynamic and inclusive. Instructors should also integrate functions into interdisciplinary projects, allowing students to apply programming skills in mathematics, physics, biology, and even social sciences, thus broadening the relevance of Python in education.

In conclusion, teaching functions in Python should be approached as a multidimensional educational task. It requires a combination of clear theoretical explanations, practical applications, collaborative activities, and technological tools. By adopting innovative and student-centered methodologies, educators can transform the teaching of functions into an effective strategy for developing programming competence, logical reasoning, and lifelong learning skills. Ultimately, functions should be taught not only as a coding tool but also as a conceptual framework that equips learners to think like computer scientists and adapt to future challenges in the digital world.

References

1. Lutz, M. (2013). *Learning Python*. O'Reilly Media.
2. Downey, A. (2016). *Think Python: How to Think Like a Computer Scientist*. Green Tea Press.
3. Wing, J. M. (2006). "Computational Thinking." *Communications of the ACM*, 49(3), 33–35.
4. Gomes, A., & Mendes, A. (2007). "Learning to Program – Difficulties and Solutions." *International Conference on Engineering Education*.
5. Guzdial, M., & Ericson, B. (2014). *Introduction to Computing and Programming in Python: A Multimedia Approach*. Pearson.