ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #45570

Teacher-Guided Project-Based Coding Practice Enhance High Level Programming Language Learning

Ms. Chaohui Ren, Auburn University
Dr. Cheryl Seals, Auburn University

Dr. Cheryl Denise Seals is a professor in Auburn University's Department of Computer Science and Software Engineering. She graduated with a B.S. C.S. from Grambling State University, M.S. C.S. from North Carolina A&T State University and a Ph.D

Karen Nix, Auburn University

Karen Nix is a PhD candidate at Auburn University, Samuel Ginn College of Engineering. She received a bachelor's degree in Computer Science from LaGrange College and a master's degree in Computer Science with a concentration of Software Development from Columbus State University. She works as a Graduate Teaching Assistant at Auburn University and recently began work for the City of Opelika, AL in the IT department as Assistant CIO. Her research interests include digital learning, UI/UX, web development, cybersecurity, web accessibility.

Teacher-Guided Project-Based Coding Practice Enhances High-Level Programming Language Learning

Abstract

Java is a required course for many undergraduate computer science programs and is widely regarded as a challenging programming language due to its complexity and abstract concepts. This paper introduces a teacher-guided, project-based coding practice designed to enhance students' understanding of Java and support their learning of high-level programming concepts. The proposed method combines teacher-led demonstrations with hands-on projects, bridging theoretical knowledge with practical applications. We conducted a case study using the *LinkedSet* project to illustrate the effectiveness of this approach. Furthermore, comparative data analysis between semesters with and without teacher-guided interventions is included to validate the impact on student performance. This study highlights how structured, interactive teaching methods can address common challenges in learning advanced programming languages and improve educational outcomes.

1 Introduction

Java is a critical component of computer science curricula due to its relevance in industry and academia. However, its steep learning curve poses significant challenges for students and instructors alike. Traditional lecture-only teaching methods often fall short in helping students grasp abstract concepts such as object-oriented programming (OOP), polymorphism, and generics.

To address these challenges, we propose a teacher-guided, project-based coding practice. This approach supplements traditional lectures with interactive lab sessions where students actively engage with real-world projects under instructor guidance. By contextualizing theoretical knowledge through practice, this method bridges the gap between theory and application, fostering deeper understanding and skill development.

2 Challenges in Teaching and Learning Java

Java's complexity stems from its rigid syntax, abstract OOP principles, and vast ecosystem of APIs and libraries. These factors contribute to difficulties in:

- Understanding core concepts such as inheritance, interfaces, and polymorphism.
- Configuring development environments and debugging errors.
- Applying theoretical knowledge to practical programming tasks.

While these challenges are well-documented [1, 2], our approach addresses them by integrating hands-on projects that align with lecture content and leveraging teacher-guided sessions to provide immediate feedback and support.

3 Methodology

3.1 Teacher-Guided Project-Based Coding Practice

To address the challenges of teaching and learning Java, we developed a structured methodology that integrates theoretical and practical components. This approach is designed to enhance student engagement, reinforce conceptual understanding, and bridge the gap between theory and application. To address the limitations of solely relying on teacher-guided instruction, we designed a learning cycle that promotes active participation and sustained engagement. This cycle leverages flipped classroom practices and periodic quizzes to ensure students do not become overly reliant on guided coding sessions. Our methodology focuses on encouraging students to take ownership of their learning while maintaining a structured and supportive environment.

Our methodology involves four key elements:

1. **Structured Lab Sessions:** Students participate in 50-minute lab sessions that complement lecture topics.
2. **Project-Based Learning:** Projects such as *LinkedSet* are designed to reinforce concepts like interfaces, generics, and OOP.
3. **Interactive Guidance:** Instructors demonstrate key coding techniques and guide students through implementation challenges.
4. **Learning Cycles:** To maintain student engagement and balance stress levels in teacher-guided project-based labs, we implemented a learning cycle mechanism. This cycle consists of teacher-guided learning, self-paced code review, flipped classroom practices [3], and quiz-based reinforcement.

   The flipped model encourages students to take an active role in their learning by requiring them to present a specific topic in the next session. During their presentation, students explain their approach, demonstrate their code, and justify their choices. This role reversal, where students become teachers, is grounded in researches [4–6]suggesting that teaching a concept significantly enhances understanding and motivation.

In addition, quizzes are incorporated to maintain a level of productive pressure, ensuring students remain motivated. The combination of flipped learning and quiz practice fosters active participation and reflection, leading to deeper comprehension and long-term retention. By engaging students in this learning cycle, we maximize their involvement, encourage peer interaction, and enable new learning and thinking to occur.

3.2 Case Study: Using the LinkedSet Project to Teach Java Concepts

3.2.1 Purpose

The LinkedSet project is a practical exercise designed to teach key programming concepts by having students implement a custom set collection. This project emphasizes the application of foundational Java topics, such as interfaces, generics, iterators, and data structures, in a real-world coding scenario. By bridging theoretical knowledge with hands-on implementation, the project reduces the intimidation of abstract concepts and fosters deeper engagement.

3.2.2 Teaching Approach

Introduction of Theoretical Knowledge: Before beginning the project, students are introduced to the Java Collections Framework and its hierarchy (See Figure 1 was adapted from [7].). For example:

- The `Collection` interface is the root of the Java Collections Framework and provides general methods for handling groups of objects.
- The `Set` interface extends `Collection`, inheriting its methods and adding its own unique property: sets cannot contain duplicate elements.
- This hierarchy illustrates how general concepts in programming (like collections) can be specialized for specific use cases (like sets).

Project-Based Application: Students are tasked with implementing `LinkedSet`, which adheres to the `Set` interface while using a doubly-linked list for storage. Key Java concepts applied include:

- **Generics:** The use of `<T extends Comparable<T>>` enforces type safety and enables comparisons for maintaining order.
- **Iterators:** Custom iterators for ascending and descending order traversal help students understand the `Iterator` interface.

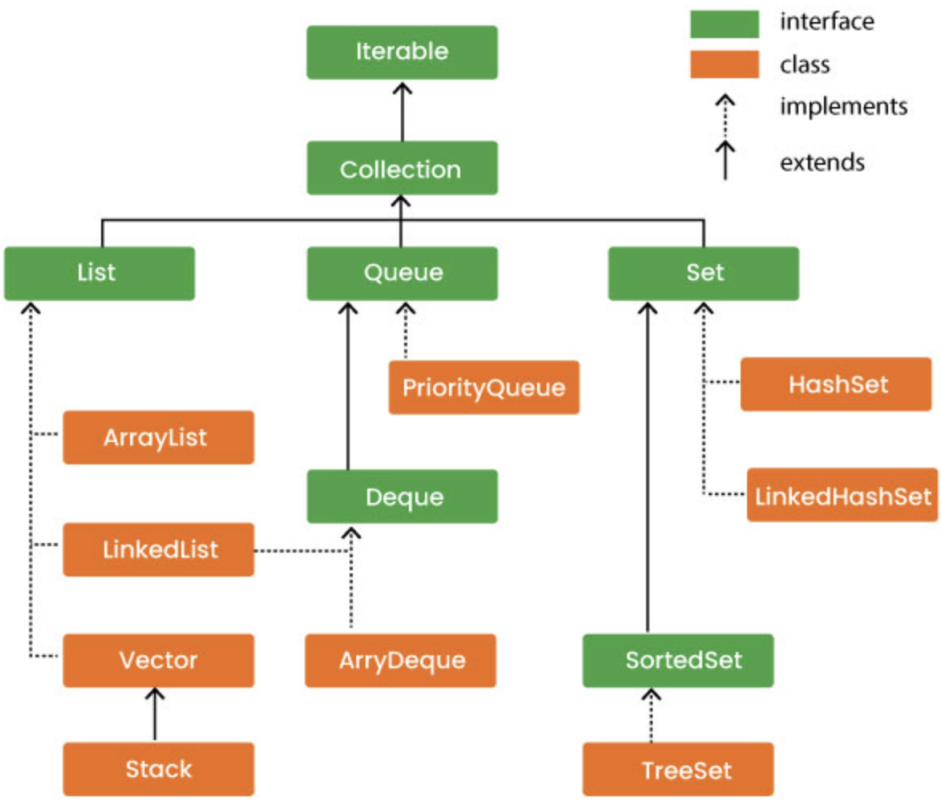

Figure 1: Overview of Java Collections Framework [7].

- **Polymorphism:** By implementing the `Set` interface, students see how concrete classes can provide specific behaviors while adhering to abstract definitions.

Guided Exploration: Teacher-guided sessions focus on challenging parts of the project, such as maintaining ascending order in the doubly-linked list when adding or removing elements. Instructors:

- Use diagrams to visualize the doubly-linked list structure (See Figure 2.), helping students grasp its relationships, invite students to draw their own boubly-linked list.
- Discuss edge cases (e.g., adding elements to an empty list or removing the front/rear node) to develop critical thinking.
- Encourage students to annotate their code with reflective comments, such as:

  "Why is reconnecting the front or rear node simpler than a middle node? Draw a graph to demonstrate."

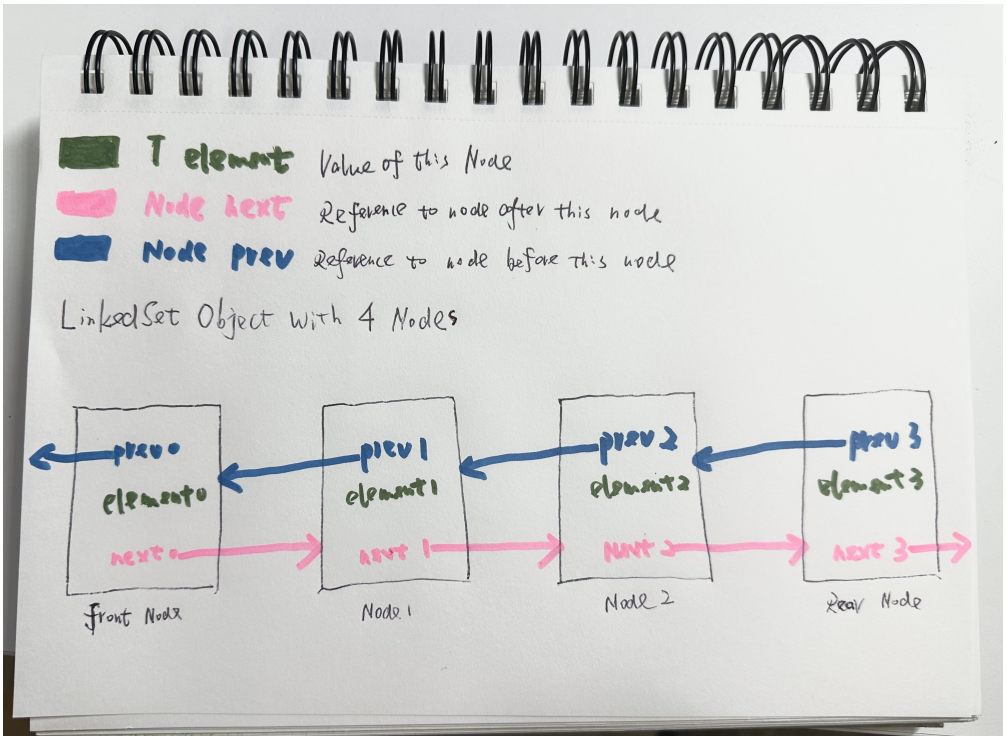

Figure 2: LinkedSet Object Example

3.2.3 Flipped Classroom Component

After completing portions of the project, students present specific methods (e.g., `add`, `remove`) in class. During presentations, students:

- Explain their implementation process and decisions.
- Demonstrate their code's functionality.
- Address questions from peers, fostering interactive learning.

This flipped model empowers students to take ownership of their learning, deepening their understanding through teaching [4–6].

3.2.4 Learning Outcomes

The LinkedSet project delivers the following benefits:

- **Improved Understanding of Abstract Concepts:** Students grasp complex ideas like generics and iterators by implementing them in a concrete context.

- **Active Learning:** The immediate application of theory to a real project increases engagement and motivation.
- **Critical Thinking:** Analyzing edge cases and debugging reinforces problem-solving skills.

3.2.5 Relevance Beyond Java

While the project focuses on Java, the skills learned—such as designing data structures, applying abstract concepts, and solving real-world coding problems—are broadly transferable to other programming languages and domains.

3.2.6 Summary of the Case Study

The LinkedSet project exemplifies the effectiveness of project-based learning in programming education. By combining teacher guidance, active student participation, and immediate application of theory, it addresses common challenges in Java learning while fostering critical programming skills.

3.3 Data Collection and Analysis

To evaluate the effectiveness of the teacher-guided method, we collected and analyzed retrospective data from two semesters: Fall 2023 (traditional lecture-only approach) and Spring 2024 (teacher-guided approach). The exams for both semesters were identical, ensuring consistency in assessment, and the data was downloaded from Canvas without any personally identifiable information, thus avoiding ethical concerns.

3.3.1 Data Selection and Metrics

Exam questions related to the LinkedSet project were manually identified based on their alignment with the concepts taught during the project. These questions assessed students' understanding of core Java topics such as:

- Interfaces and generics.
- Iterators and data structures.
- Polymorphism and object-oriented principles.
- Set operations and structural manipulation in collections.

The following metrics were analyzed for each selected question:

- **Correct Student Ratio:** The percentage of students who answered correctly.
- **Difficulty Index:** A measure of how challenging the question was for students.
- **Variance:** Reflecting the distribution of scores to understand consistency among student performance.

3.3.2 Comparative Results

Metric	2023 (Traditional)	2024 (Teacher-Guided)	Change	Observation
Correct Student Ratio (Average)	78%	75%	-3%	Slight decline.
Difficulty Index (Average)	0.65	0.7	+0.05	Students found questions slightly harder.
Variance (Average)	0.15	0.18	+0.03	Increased variability in performance.

Table 1: Comparison of Key Metrics for LinkedSet-Related Questions

3.3.3 Insights and Observations

The data reveals mixed results:

- **Correct Student Ratio:** The slight decline suggests that while the teacher-guided method benefited some students, others may have struggled with the transition or lacked sufficient engagement.
- **Difficulty Index:** The slight increase indicates that students found the questions more challenging, potentially reflecting the need for better alignment between lab sessions and exam preparation.
- **Variance:** The increased variability highlights a growing gap between high-performing and low-performing students, emphasizing the need for tailored interventions.

3.3.4 Summary

The analysis underscores the importance of optimizing lab session attendance and ensuring that all students can fully benefit from the teacher-guided method. While the method shows promise in engaging students and deepening their understanding, further refinements are necessary to achieve consistent improvements across the board.

4 Reflection and Improvements

The comparative analysis reveals that while the teacher-guided method introduced a more engaging and interactive learning environment, its initial implementation faced challenges:

4.1 Potential Factors for Decline in Performance

- **Student Adaptation:** The active learning approach required students to take greater responsibility for their learning. Some students may have struggled to adapt, particularly those accustomed to traditional lecture formats.
- **Lab Session Attendance:** A key challenge in both 2023 and 2024 was the non-mandatory nature of the lab sessions, which were treated as optional teaching assistant office hours. Observations indicate that attendance was consistently low, with only 2-3 students typically participating. Before holiday periods, attendance often dropped to zero. This limited the reach and impact of the teacher-guided method.
- **Project Complexity:** The LinkedSet project, while effective for teaching key concepts, may have been too demanding for students with weaker foundational skills.
- **Instructional Balance:** The time allocated for teacher-guided sessions and self-paced learning may not have been sufficient to fully reinforce concepts.

4.2 Proposed Improvements

To address these challenges, we propose the following refinements to the teaching methodology:

- **Mandatory Participation Policy:** Incorporate lab sessions into the formal course structure, with minimum attendance requirements or incentives such as bonus points or certifications for completion.
- **Online Access to Lab Content:** Provide recorded lab sessions or asynchronous materials to ensure that all students have access, even if they are unable to attend in person.
- **Incremental Projects:** Introduce smaller, incremental projects leading up to the LinkedSet project to build foundational skills progressively.
- **Enhanced Feedback Mechanisms:** Provide more frequent and detailed feedback during teacher-guided sessions to support struggling students.
- **Optimized Timing for Lab Sessions:** Reevaluate the timing of lab sessions to align better with student schedules and ensure higher attendance. Consider concentrated lab sessions during high-need periods such as midterms and finals.

- **Student-Centered Adjustments:** Conduct periodic surveys to gather student feedback and adjust the teaching approach to better meet their needs.

4.3 Positive Feedback from Participating Students

Despite the challenges, students who regularly attended the lab sessions in Spring 2024 provided highly positive feedback regarding the teacher-guided approach. These students reported that the sessions significantly enhanced their understanding of key Java concepts, particularly in areas such as generics, iterators, and data structures.

They also noted that the project-based teaching method allowed them to engage with abstract concepts more closely by directly applying them in code and running the implementations to see tangible results. This hands-on process not only deepened their comprehension but also provided a sense of achievement that motivated further participation. Compared to traditional lectures, where abstract concepts often felt confusing and discouraging, the lab sessions made learning more concrete and rewarding. One student remarked:

> *"The lab sessions helped me understand concepts that seemed impossible during lectures. Applying these ideas to code and seeing my implementation work gave me a real sense of accomplishment."*

This feedback highlights the potential of the teacher-guided method to improve learning outcomes when students actively participate. It underscores the importance of increasing student engagement and attendance to fully realize the benefits of this teaching approach.

4.4 Long-Term Strategies

Future iterations of this teaching approach will include longitudinal studies to track student performance over multiple semesters, enabling a more comprehensive evaluation of its impact. Additionally, incorporating peer mentoring programs may help bridge the gap for students requiring additional support.

5 Discussion

The analysis of retrospective data provides valuable insights into the effectiveness of the teacher-guided, project-based coding practice. While the method showed promise in fostering engagement and deepening students' understanding of complex Java concepts, mixed performance results reveal areas that require attention.

A notable challenge was the non-mandatory nature of lab sessions, which limited the reach of the intervention. Students who attended the sessions regularly reported significant benefits, including

improved comprehension of abstract concepts such as generics and iterators, as well as increased confidence in applying theoretical knowledge to practical tasks. However, low attendance rates meant that many students did not experience these benefits.

The increased variability in performance highlights the need for tailored support to bridge the gap between high-performing and struggling students. Additionally, the slightly higher difficulty index for exam questions suggests that better alignment between lab sessions and assessments could help mitigate these challenges. Addressing these issues through mandatory participation policies, incremental project designs, and enhanced feedback mechanisms will be critical to refining the teaching methodology.

6 Conclusion and Future Work

The teacher-guided, project-based coding practice effectively bridges the gap between theoretical instruction and practical application, offering students a hands-on approach to mastering complex programming concepts. The case study demonstrated that students benefited from the structured lab sessions, which facilitated active learning and deeper engagement. The flipped classroom model further reinforced comprehension by encouraging students to take ownership of their learning.

However, the comparative data analysis revealed mixed outcomes. While some students thrived under the teacher-guided method, others struggled to adapt, particularly due to low lab attendance and the demanding nature of the project. These findings underscore the need for refinements, such as mandatory lab participation, incremental project scaffolding, and improved alignment between teaching methods and assessments.

Future work will focus on expanding the scope of this approach to include multiple institutions and diverse student populations. Longitudinal studies will be conducted to assess the long-term impact of the methodology on programming proficiency. Additionally, integrating peer mentoring programs and leveraging technology to provide asynchronous access to lab content will further enhance the scalability and effectiveness of the approach. These efforts aim to establish a robust, student-centered model for teaching advanced programming languages.

References

[1] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," *ACM SIGCSE Bulletin*, vol. 37, no. 3, pp. 14–18, 2005.

[2] C. Watson and F. W. B. Li, "Failure rates in introductory programming revisited," in *Proceedings of the 2014 conference on Innovation and Technology in Computer Science Education*. ACM, 2014, pp. 39–44.

[3] R. Talbert, *Flipped Learning: A Guide for Higher Education Faculty*, 1st ed. Routledge, 2017. [Online]. Available: https://doi.org/10.4324/9781003444848

[4] J. A. Bargh and Y. Schul, "On the cognitive benefits of teaching," *Journal of Educational Psychology*, vol. 72, no. 5, pp. 593–604, 1980.

[5] R. D. Roscoe and M. T. H. Chi, "Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions," *Review of Educational Research*, vol. 77, no. 4, pp. 534–574, 2007.

[6] L. Fiorella and R. E. Mayer, "The relative benefits of learning by teaching and teaching expectancy," *Contemporary Educational Psychology*, vol. 38, no. 4, pp. 281–288, 2013.

[7] T. Point, "Java Collections Framework - Tutorial," 2025, accessed: January 25, 2025. [Online]. Available: https://www.tutorialspoint.com/java/java_collectaions.htm