Facilitating Student's Learning Transfer in a Database Programming Class

Yuzhe Zhou
Purdue University
West Lafayette, IN, USA
zhou922@purdue.edu

Alejandra J. Magana
Purdue University
West Lafayette, IN, USA
admagana@purdue.edu

Tianyi Li
Purdue University
West Lafayette, IN, USA
li4251@purdue.edu

Abstract

Transferring programming skills learned in the classroom to diverse real-world scenarios is both essential and challenging in computing education. This experience report describes an approach to facilitate learning transfer by fostering adaptive expertise. Students were engaged in co-creating contextualized worked-out examples, including step-by-step solutions. Through three homework assignments in a Spring 2023 database programming course, we observed substantial improvements, where students generated detailed and accurate solutions and enriched their problem-solving contexts from simple phrases to detailed stories, drawn from 17 real-life scenarios. Our results also suggest that the peer assessment process cultivated a supportive learning environment and fostered adaptive expertise. We discuss the lessons learned and draw pedagogical implications for integrating student-generated contextualized materials in other programming courses.

CCS Concepts

• **Applied computing** → **Education**.

Keywords

Learning Transfer, Adaptive Expertise, Programming, Contextualization, Self-Explanation

ACM Reference Format:
Yuzhe Zhou, Alejandra J. Magana, and Tianyi Li. 2025. Facilitating Student's Learning Transfer in a Database Programming Class. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE TS 2025), February 26-March 1, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3641554.3701828

1 Introduction

As computer science (CS) continues to evolve and intersect with various disciplines, the ability to apply theoretical CS knowledge and skills across diverse contexts – learning transfer [5, 28] – also becomes increasingly crucial. Learning transfer enhances deep understanding and critical problem-solving abilities, ensuring that students not only memorize concepts but truly comprehend and effectively utilize them in various real-world applications.

SIGCSE TS 2025, February 26-March 1, 2025, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0531-1/25/02
https://doi.org/10.1145/3641554.3701828

Researchers have explored strategies to improve students' learning transfer. Approaches like pair programming [19] and contextualization [6, 10] aim to develop adaptive expertise [9] – the ability to apply knowledge creatively to novel problems. Despite their potential, these methods rely on instructor-generated materials, which may not scale well with students' diverse and changing interests.

In this work, we explore engaging students in creating their own contextualized learning materials. Through the lens of adaptive expertise [15], we hypothesize that students would actively seek connections they could build between computing concepts and their familiar contexts (*innovation*). Additionally, the process of developing step-by-step solutions will promote self-explanation [18], thereby enhancing problem-solving skills[16] (*efficiency*).

This experience report proposes an innovative learning activity for computer science education to facilitate students' learning transfer and problem-solving skills. We categorize four different adaptive expertise development trajectories based on students' performance changes across homework assignments within a semester: Top Performer, Mountain Climber, Rollercoaster, and Slope Slider. We also draw implications for pedagogy and learning design in computing education to support more adaptable, engaging, and supportive learning environments.

2 Related Work

Learning transfer refers to an individual's ability to apply acquired knowledge in new contexts. Different learning theories provide varying perspectives on learning transfer. Behaviorism views learning transfer as a generalization, where previously learned behaviors are applied to new but similar situations [5, 28]. Cognitivism asserts that learning transfer depends on how information is stored in memory and occurs when learners apply knowledge in different contexts [5]. Constructivism suggests that learning transfer requires engaging students in authentic tasks and meaningful contexts [5].

In computer science education, various methods guided by different principles, such as problem-based learning [1], inquiry-based learning [8], and cooperative learning [21], help facilitate learning transfer. These methods involve instructors acting as facilitators and students identifying their knowledge gaps and skills to develop. Community support through group work is crucial, fostering critical analysis and problem-solving abilities [30].

Structured Query Language (SQL) is a foundational topic in computer science education that has garnered significant research attention within the CSE community [3, 12, 14, 29]. One major challenge for novices in learning SQL is understanding error messages from relational database management systems (RDBMS). Researchers have addressed this issue by developing tools like SQL-Tutor [13] and SQLValidator [14]. Researchers also tried to introduce new education methods into the field of SQL education. For example, gamification has been widely applied to increase student engagement, motivation, and performance [20, 22, 24, 25].

While these learning systems and educational approaches have shown their effectiveness, there is an untapped opportunity to enhance their impact by incorporating more varied and dynamic practice contexts that are customized to individual students' familiar scenarios and lived experiences. In this report, we propose an innovative learning activity to facilitate learning transfer and develop adaptive expertise in computer science education. We present our experience in applying it to a database programming class focused on PL/SQL.

3 Pedagogical Framework

This work views the programming learning process through the lens of the adaptive expertise framework [4]. This framework models the learning process as the development of two types of expertise, routine expertise and adaptive expertise (Figure 1). Routine expertise focuses on efficiency and accuracy, such as answering multiple-choice questions and/or programming prompts within a given amount of time. Adaptive expertise involves applying knowledge appropriately and creatively in novel situations.

Fostering one's adaptive expertise is crucial for learning transfer as it requires individuals to adapt their skills to various contexts and challenges [11, 16, 17]. The importance of adaptive expertise has been highlighted in diverse education settings [11, 16, 17].

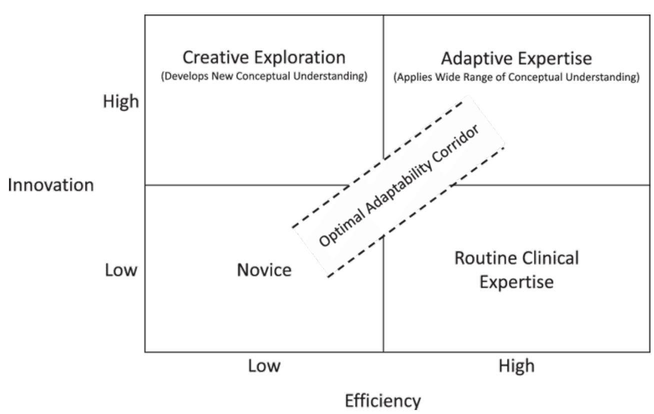


Figure 1: The Adaptive Expertise Framework portrays the learning stage with two dimensions: innovation and efficiency. [4]

3.1 Learning Objectives

Corresponding to the two dimensions (innovation and efficiency) in the adaptive expertise framework, we focus on the following two learning objectives:

Objective 1: Innovation. Identify the diverse scenarios to apply the knowledge learned from the classroom.

Objective 2: Efficiency. Decompose the process of applying learned knowledge to solve a real-world problem.

3.2 Learning Activity

We design a learning activity where students identify and formulate real-world scenarios to apply the database concepts (objective 1) and create worked-out examples with step-by-step solutions (objective 2). Worked-out examples are a group of step-by-step illustrations of the process required to complete a task or solve a problem that contains a problem context, a procedure for solving the problem, and auxiliary representations of a given problem [26].

The learning activity contains two major components. The first component focuses on routine expertise. In our activity design, this component has two practice questions related to the topic in focus. These questions are designed by the instructor to remind students of the knowledge learned from the lecture. The second component emphasizes adaptive expertise and guides students to create worked-out examples through four phases. In phase 1, students ideate and identify real-world scenarios in which the topic can be applied. In phase 2, students formulate a database problem within the scenario and define a problem statement. In phase 3, students build data tables and generate dummy data to represent example cases of the problem. These steps are designed to guide students to contextualize newly learned knowledge in real-world problems from their familiar contexts (*Learning Objective 1*). In phase 4, students suggest a solution to the problem and break the solution down into multiple smaller steps. In each step, they define the subgoal [7] and the partial solutions.

3.3 Grading Rubrics

We assess **Objective 1: Innovation** based on the problem contexts created by the students. A student's performance is considered as high in innovation if the problem context reflects their own interest or lived experience; and conversely, innovation is rated as low when the student's problem context closely mirrors the instructor's examples.

Objective 2: Efficiency is evaluated based on the step-by-step solutions created by students. A student demonstrates high efficiency if they correctly apply the given concept with an accurate and logical step-by-step solution. Conversely, if the solution contains major errors, the student's performance is rated as low in efficiency.

Student performance across both dimensions is then mapped to each quadrant in Figure 1.

4 Methods

We incorporated the learning activity into a 16-week database programming class during the Spring 2023 term. The research was approved by the IRB at the authors' institution.

4.1 Course Context

This course covers three main topics about PL/SQL, a procedural language extension for SQL (Structured Query Language): (1) PL/SQL basics (block structure, variables, data types, control structures (loops and conditional statements), SQL integration, and exception handling; (2) Functions and procedures (are schema objects that encapsulate SQL and PL/SQL statements for specific tasks); (3) Triggers (are special types of stored programs that automatically execute in response to certain events, such as INSERT, UPDATE, or DELETE operations on a database table.)

4.2 Participants

There were 23 students enrolled in this database programming class during Spring 2023 semester. The students were primarily juniors or seniors majoring in information technology or related fields. They enrolled in the course to fulfill their major requirements and all of them had completed an introductory-level SQL course. The class is offered every semester, with a class size of 20-25 in the Spring and 40-50 in the Fall with a female-to-male ratio of 1:4, which is consistent with the college's demographic.

4.3 Procedures

We implemented our designed learning activity through three homework assignments on PL/SQL basics, functions and procedures, and triggers. These assignments were given one to two weeks after the topics were introduced in the lectures and practiced in lab sessions, with students having about two weeks to complete each one. Submissions were collected via the university-licensed version of Qualtrics [1]. Each assignment accounted for 4% of the final grade and the grade was determined based on effort.

After the due date for each homework assignment, students participated in anonymous peer assessments. Reviewers were tasked with working through the examples created by their peers. They evaluated the problem context, descriptions, and data and compared their own solutions to the provided ones. Additionally, they were encouraged to leave constructive comments.

4.4 Data Analysis Methods

Each student was de-identified with a random, unique ID (S<#>). Each submission is identified as S<#>-HW<#>. All analyses were conducted after course grades were finalized, and the analysis ratings had no impact on the grades received by the students.

For the innovation assessment, two authors independently categorized the themes of problem contexts from a third of the submissions. Through iterative discussions, they refined these categories. One author then applied the refined categories to the remaining submissions, with all authors reviewing and consolidating the final themes across all submissions.

For the efficiency assessment, one author evaluated the solutions by identifying syntax and logic errors. To ensure consistency, a second author sampled 10 submissions for double-checking. This process continued until no disagreements were identified.

5 Evaluation Results

We collected 65 student-generated worked-out examples through three homework assignments, with one submission missing in the first homework and two in the second. 61 submissions received at least one peer assessment. However, one submission from the second homework and three from the third were late and did not receive peer assessments.

We classified the individual performance change with the following categories (Figure 2): **Top Performer** refers to a student who

Development Trajectories	Student's Performance Change in Single Dimension	Innovation (Number of Students)	Efficiency (Number of Students)	Both Dimensions (Number of Students)
Top Performer	↑ ↑ ↑	9	11	6
Mountain Climber	↓ ↓ ↑	2	0	0
	↓ ↑ ↑	2	6	2
Rollercoaster	↑ ↓ ↑	2	1	1
	↓ ↑ ↓	0	1	0
Slope Slider	↑ ↑ ↓	4	0	0
	↑ ↓ ↓	0	0	0

Figure 2: Single Dimensional Learning Patterns with Number of Students in Each Dimension. No student submissions were rated as low innovation or low efficiency in all three homework.

maintains a high level of innovation or efficiency. Such students received a rating of high for the dimension across all three homework assignments. **Mountain Climber** refers to a student who initially had a low level of innovation or efficiency but showed improvement as the semester progressed and finally received a rating of a high level of innovation or efficiency. **Rollercoaster** refers to a student whose level of innovation or efficiency went up and down across the three homework assignments. **Slope Slider** refers to a student who initially showed a high level of innovation or efficiency but later dropped to a low level in the later homework.

5.1 Learning Objective 1: Innovation

5.1.1 Individual Student's Performance across the Three Homework. We have observed all four learning patterns in the innovation dimension at the individual level.

Nine students were **top performers** of innovation, whose worked-out examples were constantly different from the running examples by the instructor. Four of these nine students used three different contexts in their three homework submissions. We also observed that one student, S9, included the same contexts in the first two homework submissions but changed to another context in the third homework assignment.

There were four **Mountain Climbers**. Two submissions (S8 and S18) initially used contexts similar to the running example, but identified new contexts in the third assignment. S8's worked-out example focused on student information and S18's on employees. Two students' (S6 and S23) first homework submissions were rated as low innovation due to the brevity of their problem contexts. For example, S23's context only included four words - "Fruit store orders processing." However, in the subsequent homework assignments, although S23's problem context focused on the same topic, the student provided more detailed descriptions. S6 improved the descriptions of contexts in a similar way as S23, but also chose different contexts in the remaining two homework submissions.

[1] https://purdue.qualtrics.com/

Two students were **Rollercoasters**. The students performed well in the first and third homework assignments, with problem contexts focused on job search and online business. However, in the second homework assignment, the students reused contexts similar to the ones used in the lab assignments.

Four students were **Slope Sliders**. These four students all presented two different problem contexts in their first and second homework assignments, applying database concepts to problems such as time management, soccer teams, and video games. However, for the third assignment, they all submitted scenarios similar to the questions in the running example or lab assignments.

5.1.2 Class's Collective Performance in Innovation. Overall, 16 out of 22 Homework1 (HW1) submissions, 13 out of 20 Homework2 (HW2) submissions, and 17 out of 23 Homework3 (HW3) submissions were rated as highly innovative. Our thematic analysis identified 17 types of contexts from 65 student-generated worked-out examples across all homework assignments (Table 1).

The most mentioned theme of context was "student's information," which contained topics related to students' daily lives, such as time management, courses taken, and financial aid applications. One notable theme was "job hunting," which is particularly relevant to students in their junior or senior year. In addition, some students built connections between the knowledge practiced and their personal interests, such as video games, sports, and even auto racing. Besides these, one student tried to include topics related to drought in her worked-out example.

While the number of highly innovative submissions (yellow and green boxes in Figure 3) decreased from Homework1 to Homework2 and Homework3, a deeper analysis reveals a shift in the nature of innovation. Homework1 primarily explored 13 types of problem contexts (Table 1). Students introduced three new contexts in Homework2 and one more in Homework3. Furthermore, recurring contexts, such as the "car dealership" theme, were approached from novel angles. In Homework1, one student focused on selling and trading at a car dealership, while another student focused on car repair in Homework2. This trend, coupled with the significant number of students incorporating new contexts into their later submissions, suggests a broadening of innovative thinking within the course.

Furthermore, feedback from the peer review process indicated positive reactions from students. A total of 56 comments were collected from 62 peer assessments across three homework assignments. Within the peer feedback, 26 comments commended the creators for their efforts and success in integrating diverse problem contexts. Terms like "Creative/Creativity" and "Inspiration" were frequently used by student reviewers to express their appreciation. Additionally, 8 comments specifically acknowledged the examples reviewed as excellent demonstrations of applying theoretical knowledge in practical, "real-world" scenarios.

5.2 Learning Objective 2: Efficiency

5.2.1 Individual Student's Performance Change across the Three Homework. In the dimension of efficiency, more students' submissions maintained high levels for all three homework.

Eleven students were **Top Performers**, whose submissions were rated as high efficiency in all three homework. Their solutions usually contain five or more steps. Some of them even provided test cases that other students could use to test their solutions. Moreover, their descriptions in each step were clear and easy to follow.

Six students were **Mountain Climber** – demonstrating improved efficiency across three homework assignments. All of these six students started their first homework with a low level of efficiency. The main reason was they used SQL instead of PL/SQL or made mistakes in proposed solutions. All of them improved and maintained a high level for Homework2 and Homework3.

Two students were **Rollercoasters**, whose efficiency levels fluctuated in three homework assignments. These two students followed different paths. One of them displayed high levels of efficiency in the first and third homework assignments but received a low rating for the second homework assignment due to incomplete submission, where the student only completed one of the required steps in providing solutions and left the rest of the spaces blank. The other student did not break down the solution into smaller steps in Homework1 and Homework3 and received low ratings as a result.

5.2.2 Class's Collective Performance in Efficiency. To evaluate the class's overall efficiency, we analyzed the solutions submitted by the students during the peer review process, contrasting them with those proposed by the students who originally created the worked-out examples in each homework.

We categorized solutions submitted by the students during the peer review process into three groups based on the alignment between the solutions from student reviewers and creators.

(1) Reviewer solutions were correct. 44 of the 62 peer assessments were correct. Among these 44 correct solutions, 39 aligned with the solutions provided by student creators, while five student reviewers generated answers in their own way, but also correct.

(2) Reviewer solutions were incorrect. Five peer assessments contained errors in their solutions. The main errors that showed up among these five included syntax errors, incorrect column names, and the improper use of SQL instead of the required PL/SQL.

(3) Reviewers opted to leave comments instead of solutions. This usually happens when the worked-out examples under review were considered as not "solvable" by the reviewers. Student reviewers pointed out two main issues during the peer review process. The first is related to data quality, where creators either failed to provide necessary data files or provided inaccessible files. The second is vague problem descriptions, where the worked-out examples were not clearly defined and reviewers were uncertain about how to approach the problem.

5.3 Changes in Student Performance

The last column in Figure 2 presents the number of students who exhibited similar developmental trajectories in both dimensions. Among the Top Performers, six students consistently maintained high performance in both innovation and efficiency. Additionally, two Mountain Climbers improved after starting with low performance in both dimensions in Homework1. There is also one student whose progress resembled a rollercoaster in both dimensions.

Figures 3a,3b, and 3c show where the submissions by students (represented with unique IDs from 1-23 except those four students

Table 1: Contexts Identified by Student and the number of students in each context in Homeworks 1, 2, and 3.

Themes	Description of Themes	HW 1	HW 2	HW 3
Auto Racing	Relationship between age and success in auto racing	1	0	0
Customer	Customer information such as phone numbers and zip codes	2	1	3
Car Dealership	Cars selling, trading, and repairing	2	1	1
Coffee Shop	Coffee shop cost and budgets	0	1	0
Employee	Information related to employees such as income and hiring	2	1	3
Fruit Store	Best-selling fruit and seasonal variations in fruit selling	1	1	1
Global Issues	Significant global challenge affecting environments such as drought	1	1	0
Health Care	Medical records and patient appointment management	0	1	2
IT Service	Server management in a corporate IT environment	0	2	2
Inventory Management	Information related to product inventory, inventory transaction, and status	2	3	3
Job Hunting	Application for positions that align with students' majors and experience level	0	2	1
Library	Book management	2	2	2
Movie	Types of movie and favorite movie	1	0	0
Pet Care	Pet recording, pet health tracking	1	0	0
Students' Information	Information related to students in university such as majors, classes, and GPAs	3	2	3
Sports	Soccer team performance comparison	2	1	0
Video Game	Stories in video games a student played	2	1	2

(S2, S5, S14, S21) having missed submissions) located in the adaptive expertise coordinates across the three homework.

A majority of the students (N = 12) demonstrated competencies that aligned with the Adaptive Expertise quadrant in Homework1, reflecting high innovation and efficiency. The Creative Exploration quadrant was also well-represented by four students. Three students fell into the Novice category.

Homework2 revealed a notable transition. The Adaptive Expertise quadrant retained its prominence (N = 13), though with some reconfiguration in student distribution. The Creative Exploration quadrant, however, contracted to only one student. These changes suggest a collective move towards greater efficiency. The Individual trajectory was more varied. Student 19 moved from Adaptive Expertise to Creative Exploration. Only Student 23 remained in Adaptive Expertise. Students 8 and 18 shifted to the Routine Clinical Expertise quadrant. This indicates a nuanced pattern where increased efficiency for some students coincided with a dip in innovation.

Homework3 showed dynamic yet stable class performance compared to Homework2, with a few notable individual shifts. Most students in Routine Clinical Expertise progressed to Adaptive Expertise, indicating improved levels of efficiency and innovation. However, some students reverted from Adaptive Expertise to Routine Clinical Expertise, likely due to variations in understanding and competency of different topics or other personal factors influencing the amount of effort spent on different homework assignments. Overall, the Adaptive Expertise quadrant was the most populous, showing growth in students' learning and adaptability.

5.4 Students' Comments

Our qualitative analysis of student reviewers' comments during peer assessment revealed four distinct themes. Many students found their peers' worked-out examples *educational*, often noting that these examples allowed them to practice previously learned knowledge or acquire new insights. Some included comments *suggesting revisions* to the worked-out examples. These suggestions were typically for enhancing the clarity and usefulness of the examples that were already correct. There were also comments *highlighting minor issues* in the worked-out examples. Despite these issues, reviewers expressed overall appreciation for the problems presented. Finally, some comments *identified the issues that made the worked-out examples not solvable*. Reviewers frequently cited poor data quality, unclear problem descriptions, and incorrect answers as the main problems, which corresponded with our findings in the solution comments of peer assessments.

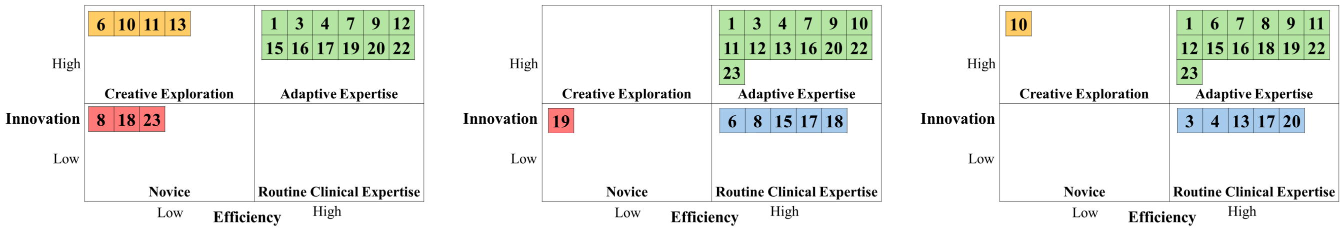
6 Discussion

Our analysis of student performance across three homework assignments in a programming course revealed how the proposed learning activities facilitated learning transfer from the development of adaptive expertise. Below we synthesize these findings and explore their broader implications for pedagogical strategies and learning design in programming education.

6.1 Learning Objective 1: Innovation

The contexts generated by students varied widely, encompassing personal interests such as sports and video games, as well as climate issues like drought. This diversity highlights not only their high level of engagement and creativity but also their ability to apply theoretical concepts to a broad spectrum of real-world scenarios. Allowing students to choose their contexts significantly enhances innovation, as it makes the learning process more relatable and engaging. This personalized approach encourages deeper cognitive processing, as students are more likely to invest effort in topics they find personally meaningful.

Moreover, the positive feedback from peer reviews highlights the value of peer assessment in reinforcing innovation. Comments praising creativity and real-world applicability suggest that peer assessment can serve as a powerful motivational tool, encouraging students to strive for innovation in their work. This peer interaction not only fostered a collaborative learning environment but also exposed students to diverse perspectives and ideas, further stimulating their creative thinking.



(a) Homework 1

(b) Homework 2

(c) Homework 3

Figure 3: Student's Adaptive Expertise in Three Homework Assignments. The numbers indicate student ID. Each colored square box represents one student and their performance standing assessed based on the adaptive expertise framework.

6.2 Learning Objective 2: Efficiency

The observation that solutions with five or more steps were generally associated with high efficiency underscores the importance of structured problem-solving and detailed explanations in developing expertise. This finding aligns with previous studies related to the self-explanation effect in learning programming identified by Pirolli and Recker [18], that students who can explain the program well, tend to perform better in programming. By breaking down problems into smaller, manageable steps, students not only demonstrated their understanding of the subject matter but also developed a more systematic approach to problem-solving.

The improvement in efficiency noted in the Mountain Climber category further illustrates the potential impact of the learning activity that requires students to articulate their thought processes. Such designs not only challenge students to clarify their understanding but also provide opportunities for reflection and self-assessment, which are key to developing efficient problem-solving strategies.

6.3 Implications for Pedagogy and Learning

The observed student's different development trajectories in two dimensions of adaptive expertise, Top Performer, Mountain Climber, Rollercoaster, and Slope Slider, underscore the need for learning environments that are adaptable and responsive to individual student needs. In our study, the fluctuations and regressions in students' performance levels in the Rollercoaster and Slope Slider patterns suggest that individual differences, such as prior knowledge, learning preferences, and perhaps external factors like workload, may significantly influence student development.

Moreover, educators should emphasize the importance of structured problem-solving and clear articulation of thought processes, especially for fields similar to programming education, which requires students to adapt theoretical knowledge and skills to solve various problems. Our findings on the effectiveness and benefit of self-explanation in learning programming align with previous studies [2, 23, 27]. Providing students with frameworks or templates for breaking down problems and encouraging them to explain their solutions in detail can help develop these essential competencies.

Finally, the steady progress made by students who approached their work like Mountain Climbers, showing continuous improvement from Homework1 to Homework3, may be attributed to the iterative nature of the assignments. Our findings on individual students' improvement in the dimension of innovation and new contexts identified from all three homework assignments indicate that creating a more engaging and supportive learning atmosphere by incorporating elements such as peer assessment could facilitate students' continuous improvement and innovation.

6.4 Limitations and Future Work

We acknowledge several limitations in our work. Firstly, students in our study come from diverse cultural, social, and academic backgrounds, which may affect their cognitive processes and the effectiveness of our learning activity, highlighting the need for thorough long-term evaluation. Secondly, as most students have completed foundational programming courses and are in their junior and senior years, this limits the generalizability of our findings. Future assessments should target a broader population. While the study took place over a semester, additional tasks like lab assignments, group projects, and potential discussions among students could contribute to enhancing student learning. Further evaluation in a controlled setting or through comparative studies is needed. Additionally, Future work should consider implementing a pre- and post-assessment to better quantify learning gains.

7 Conclusion

This study explores the development of adaptive expertise among programming students, revealing a nuanced picture of learning trajectories with notable successes and occasional setbacks. The insights offer a road map for educators seeking to cultivate high levels of innovation and efficiency in their students. By engaging students in connecting knowledge with their familiar real-world scenarios, structured problem-solving, and reflective practice, educators can better support their students on the journey toward adaptive experts in their respective fields.

Acknowledgments

This work is supported by the National Science Foundation (NSF) under grant numbers 2315683 and 2434429.

References

[1] Mohammed Abdullatif Almulla. 2020. The effectiveness of the project-based learning (PBL) approach as a way to engage students in learning. *Sage Open* 10, 3 (2020), 2158244020938702.

[2] Viviane C.O. Aureliano, Patricia C. De A.R. Tedesco, and Michael E. Caspersen. 2016. Learning programming through stepwise self-explanations. In *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*. 1–6. https://doi.org/10.1109/CISTI.2016.7521457

[3] Maurizio Cembalo, Alfredo De Santis, and Umberto Ferraro Petrillo. 2011. SAVI: a new system for advanced SQL visualization. In *Proceedings of the 2011 Conference on Information Technology Education* (West Point, New York, USA) *(SIGITE '11)*. Association for Computing Machinery, New York, NY, USA, 165–170. https://doi.org/10.1145/2047594.2047641

[4] William B Cutrer, Bonnie Miller, Martin V Pusic, George Mejicano, Rajesh S Mangrulkar, Larry D Gruppen, Richard E Hawkins, Susan E Skochelak, and Donald E Moore Jr. 2017. Fostering the development of master adaptive learners: a conceptual model to guide skill acquisition in medical education. *Academic medicine* 92, 1 (2017), 70–75.

[5] Peggy A Ertmer and Timothy J Newby. 1993. Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective. *Performance improvement quarterly* 6, 4 (1993), 50–72.

[6] Micaela Esteves, Bejamim Fonseca, Leonel Morgado, and Paulo Martins. 2008. Contextualization of programming learning: A virtual environment study. In *2008 38th Annual Frontiers in Education Conference*. F2A–17–F2A–22. https://doi.org/10.1109/FIE.2008.4720544

[7] Sebastian Garces, Camilo Vieira, Guity Ravai, and Alejandra J Magana. 2023. Engaging students in active exploration of programming worked examples. *Education and Information Technologies* 28, 3 (2023), 2869–2886.

[8] Neil Gordon and Mike Brayshaw. 2008. Inquiry based learning in computer science teaching in higher education. *Innovation in Teaching and Learning in Information and Computer Sciences* 7, 1 (2008), 22–33.

[9] Giyoo Hatano, Kayoko Inagaki, HW Stevenson, H Azuma, and K Hakuta. 1986. Child development and education in Japan. *Two Courses of Expertise* (1986), 262–272.

[10] Aleksi Lukkarinen and Juha Sorva. 2016. Classifying the tools of contextualized programming education and forms of media computation. In *Proceedings of the 16th koli calling international conference on computing education research*. 51–60.

[11] Ann McKenna, Robert Linsenmeier, and Matthew Glucksberg. 2008. Characterizing computational adaptive expertise. In *2008 Annual Conference & Exposition*. 13–288.

[12] Daphne Miedema, Efthimia Aivaloglou, and George Fletcher. 2022. Identifying SQL misconceptions of novices: Findings from a think-aloud study. *ACM Inroads* 13, 1 (2022), 52–65.

[13] Antonija Mitrovic. 1998. Learning SQL with a computerized tutor. In *Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education*. 307–311.

[14] Victor Obionwu, David Broneske, Anja Hawlitschek, Veit Köppen, and Gunter Saake. 2021. Sqlvalidator–an online student playground to learn sql. *Datenbank-Spektrum* 21 (2021), 73–81.

[15] Dana Opre. 2015. ADAPTIVE EXPERTISE: EFFICIENCY AND INNOVATION. *Cognitie, Creier, Comportament/Cognition, Brain, Behavior* 19, 2 (2015).

[16] ELİF Öztürk, Mıchael Johnson, Bugrahan Yalvac, and Xıaobo Peng. 2023. Development of Adaptive Expertise in Engineering Undergraduates through Contextual Computer Aided Design Modeling Activities. *INTERNATIONAL JOURNAL OF ENGINEERING EDUCATION* 39, 3 (2023).

[17] Marcus G Pandy, Anthony J Petrosino, Barbara A Austin, and Ronald E Barr. 2004. Assessing adaptive expertise in undergraduate biomechanics. *Journal of Engineering Education* 93, 3 (2004), 211–222.

[18] Peter Pirolli and Margaret Recker. 1994. Learning strategies and transfer in the domain of programming. *Cognition and instruction* 12, 3 (1994), 235–275.

[19] Laura Plonka, Helen Sharp, Janet Van der Linden, and Yvonne Dittrich. 2015. Knowledge transfer in pair programming: An in-depth analysis. *International journal of human-computer studies* 73 (2015), 66–78.

[20] Chanut Poondej and Thanita Lerdpornkulrat. 2020. Gamification in e-learning: A Moodle implementation and its effect on student engagement and performance. *Interactive Technology and Smart Education* 17, 1 (2020), 56–66.

[21] Sandra Schulz, Sarah Berndt, and Anja Hawlitschek. 2023. Exploring students' and lecturers' views on collaboration and cooperation in computer science courses-a qualitative analysis. *Computer Science Education* 33, 3 (2023), 318–341.

[22] Mario Soflano, Thomas M Connolly, and Thomas Hainey. 2015. An application of adaptive games-based learning based on learning style to teach SQL. *Computers & Education* 86 (2015), 192–211.

[23] Leigh Ann Sudol-DeLyser. 2015. Expression of Abstraction: Self Explanation in Code Production. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (Kansas City, Missouri, USA) *(SIGCSE '15)*. Association for Computing Machinery, New York, NY, USA, 272–277. https://doi.org/10.1145/2676723.2677222

[24] Faiza Tahir, Antonija Mitrovic, and Valerie Sotardi. 2020. Investigating the effects of gamifying SQL-Tutor. (2020).

[25] Georgi Tuparov and Daniela Tuparova. 2021. Gamification in Higher Education-a Pilot Study with SQL Course.. In *ERIS*. 81–90.

[26] Camilo Vieira, Junchao Yan, and Alejandra J. Magana. 2015. Exploring Design Characteristics of Worked Examples to Support Programming and Algorithm Design. *The Journal of Computational Science Education* 6 (July 2015), 2–15. Issue 1. https://doi.org/10.22369/issn.2153-4136/6/1/1

[27] Arto Vihavainen, Craig S. Miller, and Amber Settle. 2015. Benefits of Self-explanation in Introductory Programming. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (Kansas City, Missouri, USA) *(SIGCSE '15)*. Association for Computing Machinery, New York, NY, USA, 284–289. https://doi.org/10.1145/2676723.2677260

[28] John B Watson. 2017. *Behaviorism*. Routledge.

[29] Wensheng Wu. 2021. SQL2X: Learning SQL, NoSQL, and MapReduce via Translation. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 590–596.

[30] Enwei Xu, Wei Wang, and Qingxia Wang. 2023. The effectiveness of collaborative problem solving in promoting students' critical thinking: A meta-analysis based on empirical literature. *Humanities and Social Sciences Communications* 10, 1 (2023), 1–11.