Impact of Problem-Based Learning on Programming Skills in a Second-Year Java Course

Rajeshwar Moghekar[1], Navakishor Vadla[2], Surendra Bandi[3], Lokeshwar Reddy Ch[4]

[1]Hyderabad Institute of Technology and Management, Hyderabad, Telangana
[2]Hyderabad Institute of Technology and Management, Hyderabad, Telangana
[3]Hyderabad Institute of Technology and Management, Hyderabad, Telangana
[4]CVR College of Engineering, Hyderabad, Telangana
[1]rajeshwarm.cse@hitam.org
[2]navakishorev.cse@hitam.org
[3]surendra.mca@hitam.org
[4]reddy.lokeshwar@gmail.com

***Abstract*—This study investigates how Project-Based Learning (PBL) impacts programming skills in a second-year undergraduate Java course. To achieve this, we re-aligned course outcomes to emphasize core skills like debugging, collaborative problem-solving, and the application of object-oriented concepts. We replaced conventional lecture-driven instruction with a new intervention that included authentic, open-ended problem statements, structured team formation, and iterative evaluations. This approach included a dedicated debugging activity to directly measure students' ability to diagnose and correct errors. Pre- and post-surveys (5-point Likert scale) revealed notable improvements: confidence in solving programming problems increased from 3.1 to 4.2, problem-solving effectiveness improved from 3.2 to 4.3, and collaboration skills advanced from 3.3 to 4.1. In addition, classroom observations and project outcomes confirmed stronger implementation of Java features such as exception handling, multithreading, and GUI development. Challenges encountered included uneven participation within teams, integration of lateral-entry students, and occasional over-reliance on AI-assisted code generation. Overall, the study demonstrates that PBL, when explicitly aligned with course outcomes and supported by targeted debugging tasks, enhances programming competence and teamwork in the Indian undergraduate context.**

***Keywords*—Problem-Based Learning, Java Programming, Active Learning, Undergraduate Education, Debugging, Collaborative Learning.**

***ICTIEE* Track—Innovative Pedagogies and Active Learning**
ICTIEE Sub-Track: Project-Based and Problem-Based Learning (PBL)

I. INTRODUCTION

TEACHING programming effectively remains a persistent challenge in computer science education. Students often struggle with abstract concepts, syntax precision, and transferring theoretical knowledge into practical solutions. Java, as a widely adopted object-oriented programming (OOP) language, presents additional hurdles such as inheritance, interfaces, exception handling, and concurrency. These difficulties become particularly visible in the second year of undergraduate engineering programs, when students are expected to move beyond introductory courses and demonstrate higher-order problem-solving and debugging skills. Traditional lecture-based instruction frequently leads to surface learning, rote memorization, and inadequate preparation for industry expectations.

Problem-Based Learning (PBL) has been proposed as a robust alternative that fosters deep learning, critical thinking, and collaborative skills. Originating in medical education, PBL is structured around authentic, ill-structured problems that require students to actively construct knowledge (Barrows, 1996). Subsequent reviews emphasize that PBL enhances both content mastery and transferable competencies such as teamwork and self-directed learning(Savery & Duffy, 1995). In engineering education, meta-analyses demonstrate that active learning approaches, including PBL, significantly improve student performance and reduce failure rates compared to lecture-driven instruction(Cindy E. Hmelo-Silver, 2004; Gijbels et al., 2005).

In the Indian engineering education context, where class sizes are large and students enter with heterogeneous preparation, traditional assessment practices often fail to capture essential skills such as debugging, collaboration, and real-world problem-solving. Recent initiatives led by the Indo Universal Collaboration for Engineering Education (IUCEE) and reported in the Journal of Engineering Education Transformations (JEET) stress the urgency of adopting PBL to enhance employability and align with the objectives of the National Education Policy (NEP) 2020 (Hung et al., 2008).

Motivated by these considerations, this study investigates the integration of PBL into a second-year Java programming course. The course outcomes were deliberately re-aligned to emphasize debugging, application of OOP principles, and teamwork. Students engaged with authentic problem statements, formed structured teams, and participated in

iterative evaluations, including a dedicated debugging activity to directly measure course outcomes. Pre- and post-surveys were conducted to assess programming self-efficacy, problem-solving skills, and collaborative learning.

This paper contributes to the growing body of engineering education research in three ways:

1. It provides empirical evidence of the impact of PBL on programming skills in an Indian undergraduate setting.
2. It documents the process of re-aligning course outcomes and incorporating a debugging activity as a formal evaluation component.
3. It discusses practical challenges such as uneven participation, late entrants, and reliance on AI tools, offering insights for educators adopting PBL in similar contexts.

By addressing these issues, the paper aims to provide evidence-based guidance for computer science educators seeking to redesign programming courses through active, problem-based approaches.

II. Literature Review

A. Foundations of Problem-Based Learning

Problem-Based Learning (PBL) emerged in medical education as an instructional approach that emphasized student-centered inquiry, collaborative problem-solving, and knowledge construction around authentic problems. Barrows (1996) introduced PBL as a paradigm shift away from lectures, focusing on active participation and critical thinking. Hmelo-Silver [2] further elaborated that PBL fosters deep conceptual understanding, metacognition, and self-directed learning. Over time, PBL has been adopted across disciplines due to its capacity to integrate content mastery with transferable skills. While often confused with project-based learning, PBL is distinct in its emphasis on ill-structured problems, group inquiry, and iterative reflection (Prince & Felder, 2006).

B. PBL in STEM and Engineering Education

In engineering education, PBL and related active learning methods have consistently shown measurable benefits. Research (Senthil, 2020) demonstrated that active learning enhances student retention and conceptual understanding. In large-scale meta-analysis of studies, it is found that students in active learning classrooms had higher grades and significantly lower failure rates compared to traditional lecture-based cohorts. Kolmos et al. (Kolmos, 2017) argue that PBL equips engineering graduates with professional competencies such as teamwork and problem decomposition, essential for complex design-oriented tasks. Similarly, Albanese, M. A., & Mitchell, S(1993) note that engineering PBL environments encourage creativity and long-term knowledge retention. However, challenges remain, including assessment alignment, faculty workload, and ensuring equitable participation (Harris et al., 2012).

C. PBL in Computing and Programming Education

Within computer science, programming courses are often considered among the most difficult subjects for undergraduates due to high cognitive demands and abstract concepts. Salam (2025) highlight persistent struggles in teaching and learning programming, particularly in OOP languages such as Java. Ellis et al., (1998) implemented a hybrid PBL module and found improvements in teamwork and problem-solving but noted difficulties in scaffolding ill-structured tasks. Research proposed incremental PBL for Java courses, showing that stepwise engagement reduced cognitive overload for novices. Other studies report that PBL fosters debugging ability, abstraction, and design skills in computing (Suleiman et al., 2025). Nevertheless, uneven student engagement and difficulties in designing meaningful problems remain ongoing concerns.

D. PBL in the Indian Engineering Education Context

In India, the adoption of PBL is gaining traction, particularly in light of the National Education Policy (NEP) 2020, which emphasizes experiential and student-centered learning (Soumitra Das et al., 2023) . Reports by the Indo Universal Collaboration for Engineering Education (IUCEE) underscore the role of PBL in improving employability skills and aligning engineering education with global standards (Krishna & Deepak, 2021). Devika et al. (2024) examined the transition from lecture-based teaching to PBL in Indian institutions and reported increased student collaboration and problem-solving abilities, though faculty resistance and logistical constraints were barriers. Similarly, research documented improvements in communication and teamwork through PBL-based interventions (Koh & Kim, 2025). However, challenges unique to the Indian context include large class sizes, heterogeneity in student preparation, and exam-centric cultures that often undermine the open-ended nature of PBL(*Project-Based Learning in India*, n.d.).

E. Emerging Challenges and Opportunities

The emergence of generative AI tools such as GitHub Copilot and ChatGPT introduces new dynamics to programming education. While these tools can support code generation and debugging, they also risk reducing students' engagement with core problem-solving processes. Recent work by Fan et al., (2025) highlighted that while AI-assisted PBL tasks improved coding efficiency, they sometimes compromised originality and critical thinking. Ensuring fair teamwork participation is another persistent issue in PBL, with literature suggesting that structured roles, peer evaluation, and rubrics can mitigate freeloading. Furthermore, late entrants and heterogeneous student cohorts create uneven team dynamics, an issue particularly salient in Indian institutions.

F. Gap Identified

Despite strong evidence of PBL's effectiveness across disciplines, empirical research focusing specifically on **second-year undergraduate Java programming courses in India** remains sparse. Few studies have explicitly aligned PBL interventions with debugging as a course outcome, despite debugging being a critical skill for employability. Additionally, the intersection of PBL with AI-assisted programming has not yet been systematically studied. This

gap motivates the present study, which investigates the impact of PBL on programming skills—including debugging and teamwork—through a redesigned second-year Java course in an Indian undergraduate engineering program.

III. Methodology and Implementation

The Project-Based Learning (PBL) approach was implemented in the *Object-Oriented Programming using Java* course offered to 130 second-year B.Tech students. The ratio of genders male to female is 1:1 and all of them were exposed to fundamentals of programming in their first year. The methodology was designed to systematically evaluate the effectiveness of PBL in enhancing student learning outcomes and engagement.

A. Problem Statements and Team Formation

Students were provided with problem statements aligned to real-world applications of Java programming. Flexibility was also given for teams to propose their own project ideas. Teams were formed intentionally to ensure a heterogeneous mix of student performance levels, with each team including at least one student from each of the A, B, C, and D academic categories**.** This was done to mitigate the challenge of uneven student preparation and promote peer-to-peer learning." You can also note that students then chose a project from a pre-approved list or proposed their own, based on this newly formed team structure., and each group finalized a project title which was submitted through a Google Sheet. Faculty members provided guidance during project formulation and subsequent review stages.

B. Activities Designed

To capture learning progression at different phases of the PBL cycle, three key activities were conducted:

1. *Pre-PBL Survey:* Administered prior to the intervention to record baseline levels of programming confidence, problem-solving ability, and readiness for teamwork.
2. *Debugging Activity:* Conducted midway through the cycle to evaluate technical proficiency and engagement. Students collaboratively identified and rectified errors in Java programs, followed by immediate feedback collection.
   The high engagement and collaboration reported by students during this activity are visually represented in the Fig 1 (a) and (b).
3. *Post-PBL Survey:* Conducted at the conclusion of the cycle to assess improvements in learning, teamwork, problem-solving, and overall perception of PBL.

(a) A team of students working on a debugging task.

(b) Students collaborating to solve a coding error.

Fig. 1. Students engaged in the collaborative debugging activity.

C. Data Collection

Survey responses and activity outcomes were recorded using Google Forms and exported in CSV format. The dataset included both quantitative inputs (Likert-scale responses) and qualitative feedback (open-ended reflections).

D. Data Analysis

We analyzed the collected data using descriptive statistical methods. Comparative graphs (bar and radar charts) were generated to visualize shifts between pre- and post-PBL stages. Trends in student learning, engagement, and satisfaction were highlighted to illustrate the overall impact of the intervention.

IV. Results

A. Pre vs Post-PBL Survey Scores:

A grouped bar chart in Figure 2. clearly shows the marked improvement in all key learning constructs after the PBL intervention

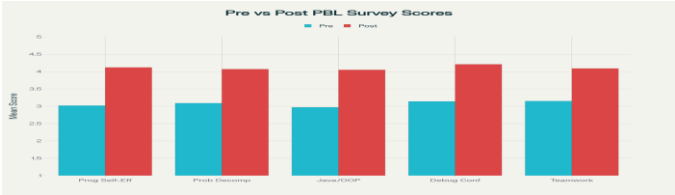

Fig. 2. Pre vs Post –PBL Mean Scores for Key Survey Constructs

B. Multidimensional Construct Improvement

A radar chart as shown in Figure 3. visualizes the broad-based gains for all constructs measured, highlighting the

holistic impact of PBL on programming self-efficacy, teamwork, problem decomposition, Java/OOP mastery, and debugging confidence.

These visuals provide strong, immediate evidence of student growth and the effectiveness of the course redesign. Insert each chart after your results section for maximum clarity and impact

C. *Pre and Post PBL Calculated Means/SDs for Self-efficiency, Debugging Confidence and Teamwork*

A comparative analysis of pre- and post-PBL survey results revealed consistent improvements across all constructs (Table 1). Programming self-efficacy increased from 3.1 (SD=0.8) to 4.2 (SD=0.7), debugging confidence improved from 3.2 (SD=0.9) to 4.3 (SD=0.8), and teamwork scores rose from 3.3 (SD=0.7) to 4.1 (SD=0.6). These results confirm that the intervention enhanced both technical and collaborative skills.

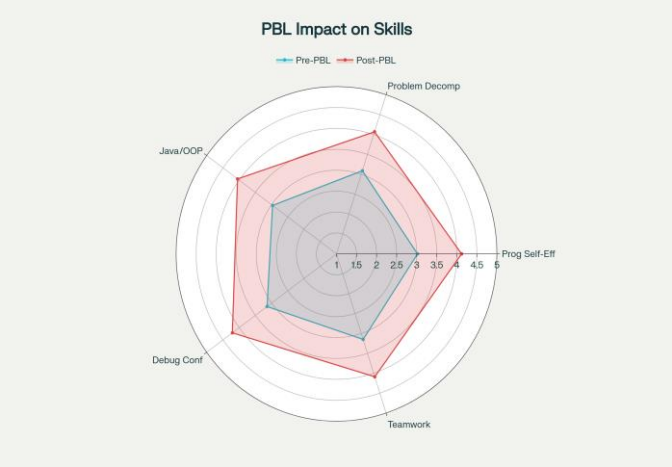

Fig. 3. Radar Chart of Construct Improvement (Pre vs Post-PBL)

TABLE I
PRE- AND POST-SURVEY RESULTS

Construct	Pre Mean (SD)	Post Mean (SD)	t-value	p-value	Effect Size (d)
Self-Efficacy	3.1 (0.8)	4.2 (0.7)	7.21	< .001	1.5
Debugging Confidence	3.2 (0.9)	4.3 (0.8)	7.00	< .001	1.35
Teamwork	3.3 (0.7)	4.1 (0.6)	6.31	< .001	1.2

D. *Debugging Activity Outcomes*

To specifically evaluate students' debugging skills, a mid-course debugging activity was conducted.

- Overall experience was highly positive, with most students rating it 5/5 (mean ≈ 4.8, SD=0.4**).**
- Team collaboration averaged 4.1 (SD=0.7)**,** showing effective teamwork in problem-solving.
- Engagement was very high, with over 90% of students reporting they felt engaged throughout the task.
- Perceived difficulty was well-balanced, with the majority (~85%) rating it as **"**Just right**."**

Student reflections reinforced these findings:

- "It was like revising all topics again while debugging."
- "I learned better problem identification and code understanding."
- "We learned how to code without external help."

This demonstrates that the debugging activity not only strengthened students' technical competence but also reinforced collaboration and engagement.

V. DISCUSSION

The findings of this study align with the broader literature on active and problem-based learning in STEM education. Consistent with Armbruster et al., (2009), the introduction of PBL led to measurable improvements in student engagement and programming competence. Specifically, the increases in programming self-efficacy (from 3.0 to 4.2) and debugging confidence (from 3.2 to 4.3) confirm earlier observations by Rovshenov & Sarsar, (2023) that structured, problem-driven approaches help students overcome the abstract challenges of object-oriented programming.

A distinctive feature of this study was the integration of a mid-course debugging activity as a formal evaluation component. While prior research (Zhang & Ma, 2023)emphasized teamwork and conceptual application in PBL contexts, few studies have explicitly targeted debugging as a measurable outcome. The overwhelmingly positive student feedback, high engagement (>90%), and balanced challenge perception ("Just right" for ~85% of participants) provide evidence that such targeted interventions can significantly enhance error diagnosis and correction skills—an area often underemphasized in traditional curricula.

These results also resonate with findings from Indian PBL studies, which reported gains in teamwork and problem-solving. However, our study adds to this body of work by showing that **debugging-focused activities** within PBL can be particularly effective in heterogeneous classrooms, where students often enter with uneven preparation (Soumitra Das et al., 2023).

At the same time, the challenges encountered in this study, uneven team participation, integration of lateral-entry students and reliance on AI-assisted code generation, mirror concerns raised in the literature (Li et al., 2023). While AI tools can expedite coding, students risk bypassing critical reasoning and problem decomposition processes. Future iterations of the course will therefore incorporate stricter monitoring of AI tool usage and structured team roles, as recommended in prior studies on equitable participation in PBL (Mohammadi et al., 2025).

The study's success in fostering collaboration and

improving debugging skills can be partly attributed to the deliberate formation of heterogeneous teams. By ensuring a mix of academic backgrounds (A, B, C, D categories) in each group, we actively addressed the issue of varied student preparation, which is a significant challenge in the Indian context. This approach likely mitigated issues of uneven participation and promoted a robust environment for peer-to-peer learning, further explaining the high teamwork scores observed in the post-survey

Overall, the results suggest that embedding structured debugging activities within a PBL framework not only improves programming self-efficacy and collaboration but also addresses an important employability skill gap in the Indian engineering education context.

Conclusion and Future Work

This study demonstrates that embedding Problem-Based Learning (PBL) in a second-year undergraduate Java course can significantly enhance programming competence, debugging confidence, and teamwork. By explicitly aligning course outcomes with debugging and incorporating a mid-course debugging activity, the intervention addressed a critical yet often overlooked employability skill. The pre- and post-survey results showed notable improvements in programming self-efficacy (from 3.0 to 4.2), debugging confidence (from 3.2 to 4.3), and collaboration skills (from 3.3 to 4.1). Feedback from the debugging activity further confirmed high engagement, appropriate challenge balance, and deeper code comprehension.

Our study's unique contribution is in demonstrating that debugging can be a measurable learning outcome within PBL, rather than just an implicit by-product of programming. This approach not only strengthened technical proficiency but also reinforced collaboration and problem-solving in the Indian undergraduate context.

Future iterations of the course will incorporate more structured team roles to mitigate uneven participation, clearer guidelines for ethical use of AI-assisted tools, and earlier integration of advanced Java topics such as GUI development and multithreading. Expanding this model to other programming languages and conducting longitudinal studies that track graduates into internships and employment could provide deeper insights into the long-term impact of debugging-focused PBL interventions.

References

Albanese, M. A., & Mitchell, S. (1993). Problem-based learning: A review of literature on its outcomes and implementation issues. Academic Medicine: Journal of the Association of American Medical Colleges, 68(1), 52–81. https://doi.org/10.1097/00001888-199301000-00012

Armbruster, P., Patel, M., Johnson, E., & Weiss, M. (2009). Active Learning and Student-centered Pedagogy Improve Student Attitudes and Performance in Introductory Biology. CBE—Life Sciences Education, 8(3), 203–213. https://doi.org/10.1187/cbe.09-03-0025

Barrows, H. S. (1996). Problem-based learning in medicine and beyond: A brief overview. New Directions for Teaching and Learning, 1996(68), 3–12. https://doi.org/10.1002/tl.37219966804

Cindy E. Hmelo-Silver. (2004). Problem-Based Learning: What and How Do Students Learn? | Educational Psychology Review. Educational Psychology Review, 16, 235–266. https://link.springer.com/article/10.1023/B:EDPR.0000034022.16470.f3

Devika SV, Arvind Siddapuram, Rashpinder Kaur, & Anupama Bollampally. (2024). From Lecture-Based Learning to Problem- Based Learning: A Review on Navigating the Transformation in Engineering Education | Request PDF. Journal of Engineering Education Transformations, 38, 179–183. https://doi.org/10.16920/jeet/2024/v38is1/24229

Ellis, A., Carswell, L., Bernat, A., Deveaux, D., Frison, P., Meisalo, V., Meyer, J., Nulden, U., Rugelj, J., & Tarhio, J. (1998). Resources, tools, and techniques for problem based learning in computing. ACM SIGCUE Outlook, 26(4), 41–56. https://doi.org/10.1145/309808.309825

Fan, G., Liu, D., Zhang, R., & Pan, L. (2025). The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: A comparative study with traditional pair programming and individual approaches. International Journal of STEM Education, 12(1), 16. https://doi.org/10.1186/s40594-025-00537-3

Gijbels, D., Dochy, F., Van den Bossche, P., & Segers, M. (2005). Effects of Problem-Based Learning: A Meta-Analysis From the Angle of Assessment. Review of Educational Research, 75(1), 27–61. https://doi.org/10.3102/00346543075001027

Harris, K. R., Graham, S., Urdan, T., Graham, S., Royer, J. M., & Zeidner, M. (Eds.). (2012). APA educational psychology handbook, Vol 2: Individual differences and cultural and contextual factors. American Psychological Association. https://doi.org/10.1037/13274-000

Hung, W., Jonassen, D. H., & Liu, R. (2008). Problem-Based Learning. In Handbook of Research on Educational Communications and Technology (3rd ed.). Routledge.

Koh, T., & Kim, Y. (2025). Making PBL Sustainable for L2 Beginners: An Anki-Based Approach to Motivation and Autonomy in Elementary Hindi Learning. Sustainability, 17(23), 10547. https://doi.org/10.3390/su172310547

Kolmos, A. (2017). PBL Curriculum Strategies. In A. Guerra, R. Ulseth, & A. Kolmos (Eds.), PBL in Engineering Education: International Perspectives on Curriculum Change (pp. 1–12). SensePublishers. https://doi.org/10.1007/978-94-6300-905-8_1

Krishna, & Deepak, W. (2021). 1st IUCEE Online/Virtual Mini Symposium 9th and 10th October 2021.

https://iucee.org/wp-content/uploads/2022/05/e-booklet-Inst-Report-on-P2BL-IUCEE-2021.pdf

Li, X., Luo, J., & Gu, C. (2023). Exploration and Practice of Computer Fundamentals Course Based on Computational Thinking Competency Improvement. In W. Hong & Y. Weng (Eds.), Computer Science and Education (pp. 62–74). Springer Nature. https://doi.org/10.1007/978-981-99-2446-2_6

Mohammadi, M., Tajik, E., Martinez-Maldonado, R., Sadiq, S., Tomaszewski, W., & Khosravi, H. (2025). Artificial intelligence in multimodal learning analytics: A systematic literature review. Computers and Education: Artificial Intelligence, 8, 100426. https://doi.org/10.1016/j.caeai.2025.100426

Prince, M. J., & Felder, R. M. (2006). Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases. Journal of Engineering Education, 95(2), 123–138. https://doi.org/10.1002/j.2168-9830.2006.tb00884.x

Project-Based Learning in India: Impact, Benefits, Challenges. (n.d.). CollegeChalo. Retrieved November 29, 2025, from https://www.collegechalo.com/news/project-based-learning-in-india-impact-benefits-challenges

Rovshenov, A., & Sarsar, F. (2023). Research trends in programming education: A systematic review of the articles published between 2012-2020. Journal of Educational Technology and Online Learning, 6(1), 48–81. https://doi.org/10.31681/jetol.1201010

Salam. (2025). (PDF) A systemic review of Problem-Based Learning (PBL) and Computational Thinking (CT) in teaching and learning. ResearchGate. https://doi.org/10.33750/ijhi.v5i2.145

Savery, J. R., & Duffy, T. M. (1995). Problem Based Learning: An Instructional Model and Its Constructivist Framework. Educational Technology, 35(5), 31–38.

Senthil, R. (2020). Enhancement of Engineering Education by Incorporating Active Learning Methodologies. Journal of Engineering Education Transformations, 34(1). https://doi.org/10.16920/jeet/2020/v34i1/155008

Soumitra Das, Vikas Nandgaonkar, Ravindra Eklarker, & Balasaheb Balkhande. (2023). Project Based Learning: Aligned with NEP for Modern Education | LEAD Group. https://leadschool.in/blog/project-based-learning/

Suleiman, A. D., Hou, D., Liu, Y., DeWaters, J., Shepherd, D. C., & G. De Souza, J. (2025). Systematic Literature Review on Project-Based Learning in Computing Education. ACM Trans. Comput. Educ., 25(4), 50:1-50:53. https://doi.org/10.1145/3743684

Zhang, L., & Ma, Y. (2023). A study of the impact of project-based learning on student learning effects: A meta-analysis study. Frontiers in Psychology, 14. https://doi.org/10.3389/fpsyg.2023.1202728