Collaborative Learning Techniques in Python Programming: A Case Study with CSE Students at Anurag University

P. Rathna Sekhar[1], Swapna Goud[2]
[1&2]Department of Computer Science and Engineering, Anurag University, Hyderabad, India.
[1]rathnasekhar.cse@anurag.edu.in [2]swapnagoudcse@anurag.edu.in

***Abstract*—Today's education system requires new, interactive methods to keep students interested and help them learn better. This paper looks at various Teaching-Learning Process (TLP) methods used to teach Python programming to B.Tech CSE students. The methods include Chalk & Talk with PowerPoint, Think Pair Share, Open-ended problems in Laboratories, Quality Circles, Concept Mapping, Web-based Learning, Moodle, Mock Interviews, and Think Aloud Pair Problem Solving. These methods were included in the curriculum to make the classroom more engaging and participative. The aim was to change traditional teaching into an interactive experience, encouraging students to actively participate, collaborate, and use their knowledge in real-life situations. Each TLP method covered different learning aspects, from understanding theory to solving practical problems. Using these methods in teaching Python showed great improvements in student involvement, understanding, and retention. These methods were used on one of the batches of CSE department, and results show a significant improvement in final results. This paper details each method, its classroom use, and the observed outcomes, offering recommendations for educators to improve teaching practices in technical education.**

Keywords: Interactive teaching methods, Teaching learning process, Pedagogy, Active learning, Python programming.

1. INTRDUCTION

Nowadays, students are heavily into technology and always connected to it. Anything involving technology grabs their attention. Integrating technology into teaching methods is an effective way to inspire today's generation to attend classes and stay engaged.

Incorporating technology into teaching methods is an effective way to inspire today's generation to attend classes and stay engaged. By using tools like interactive simulations, online collaboration platforms, and multimedia presentations, educators can create a dynamic learning environment that connects with tech-savvy students. For instance, using platforms such as Moodle for assignments and discussions or employing coding platforms for hands-on programming exercises makes learning more relatable and exciting. While incorporating technology is crucial, maintaining a balance with traditional teaching methods ensures that critical thinking and problem-solving skills are developed both with and without digital tools. This balance prepares students to utilize technology effectively while reinforcing fundamental analytical skills.

Additionally, technology supports personalized learning experiences, allowing students to progress at their own pace and according to their individual learning styles. Digital resources like educational videos, e-books, and virtual labs provide students with extra support and enrichment beyond the classroom. This approach not only improves understanding and retention but also prepares students for the technologically advanced world they will enter after graduation.

By adopting these modern teaching tools, educators can better meet their students' needs, creating a more interactive, engaging, and effective learning experience. This shift in teaching methods is essential for developing a generation of learners who are not only knowledgeable but also skilled at using technology in their problem-solving and critical thinking processes.

Research has consistently shown that active learning strategies significantly improve student engagement, comprehension, and retention of course material. Studies like those by Freeman et al. [1] have demonstrated that students in active learning environments are less likely to fail compared to those in traditional lecture-based classes (Freeman et al., 2014). Additionally, Kordaki et al. [2] found that incorporating activities such as the Jigsaw method in computer science courses leads to higher student engagement and better learning outcomes. The study made by Zuzana et al. [3] shows that implementing Inquiry-Based Science

Education (IBSE) across multiple STEM disciplines leads to a statistically significant improvement in students' inquiry skills, with an almost 8% increase. This improvement highlights the effectiveness of IBSE in enhancing students' ability to conduct scientific inquiries and develop critical thinking skills.

II. RELATED WORK

[4] explores the integration of active learning and blended methods. It demonstrates significant improvements in student outcomes, highlighting the effectiveness of discussion forums, tech talks, and quizzes, using logistic regression to validate the results. This study offers a valuable approach to modernizing software engineering education. Preeti S. Joshi et al, [5] demonstrates that animated videos and crossword puzzles significantly enhance student learning and engagement in engineering physics. Over 90% of students reported improved understanding and retention, highlighting the effectiveness of these interactive teaching methods in creating an enjoyable classroom experience. The paper "Online Teaching-Learning Experience During COVID-19 Pandemic – A Case Report" by Sandeep U. Mane, Satyajit R. Patil, and Sushma S. Kulkarni [6] explores the transition to online education during the pandemic. It finds that while 48.08% of students benefited from virtual classes, challenges such as technical issues and the lack of physical classroom experience affected overall satisfaction. The authors suggest a hybrid approach for future education. Emmanuel James P. Pattaguan [7] evaluates the Project READS program at the University of Saint Louis, Philippines. It finds that the program effectively engages engineering faculty in professional reading, enhancing their teaching and promoting a culture of continuous learning. The program is deemed effective and sustainable, with suggestions for further improvements provided.

The paper by T. Sujithra and N.M. Masoodhu Banu [8] explores the use of the Connexion game to teach C programming. The study found that 90% of students actively participated and understood the concepts better, demonstrating the game's effectiveness in making learning engaging and enjoyable. This innovative approach bridges the gap between traditional teaching methods and the interactive learning preferences of modern students. Julius Fusic S [9] explores the effectiveness of using Google Classroom for online assignment submissions. It found that this approach enhances student engagement, skill development, and timely submission. The study highlights improved learning outcomes, with a significant portion of students benefiting from the interactive and accessible nature of online assignments, making it a valuable tool for modern education. Muhammad Nurtanto, Herminarto Sofyan, and Pardjono [10] explores the development of interactive multimedia for AutoCAD 3D learning. It finds that this e-learning tool significantly enhances student engagement and learning outcomes in vocational education. The multimedia was rated "Very Good" by experts and students, making it a promising approach for technical education.

III. TEACHING METHODOLOGIES

A. Think pair share

Think: Students are given 7-10 minutes of time to think individually about what they already know about data types (like integers, floats, strings, lists, etc.) and operators (arithmetic, assignment, comparison, logical, etc.). Students are encouraged to review previous knowledge that they have with these concepts. To guide thinking prompt questions are provided like How do you think Python handles different types of data?

Pair: After allowing to think students are paired together with varying levels like high , moderate, low. Students discuss their thoughts and share what they have identified about data types and operators. Misunderstandings are clarified and concepts are deepen through conversation.

Share: Few pairs are invited to share their insights, observations and key conclusions. Students are encouraged to listen actively and ask questions. As the discussion progresses if any misconceptions are there faculty clarifies and provide additional examples and summarizes the key points.

Benefits

1. Active engagement, Collaborative learning, Deeper understanding, Increased participation, Feedback opportunity.

B. Quality circles:

Python's libraries such as Pandas and NumPy facilitate comprehensive data analysis, enabling the identification of trends and anomalies crucial for quality improvement initiatives. Through machine learning frameworks like scikit-learn, Python empowers predictive analytics, anticipating potential issues before they impact product quality.

Its integration with visualization tools like Matplotlib and Plotly allows for clear and concise data presentation, enhancing communication of findings to stakeholders. Python's simplicity and versatility make it accessible for cross-disciplinary collaboration within quality circles, fostering innovative solutions and continuous improvement strategies. This paper aims to showcase Python's transformative impact on quality management through practical examples and case studies, highlighting its efficacy in modern educational and industrial contexts.

Benefits:

1. Python simplifies complex data analysis tasks in quality circles, improving accuracy and efficiency in identifying quality issues.
2. Its integration with machine learning facilitates predictive analytics, enabling proactive quality management and defect prevention.
3. Python's visualization capabilities enhance communication of quality insights, aiding in decision-making processes and stakeholder engagement.

TABLE 1
COURSE LEARNING OUTCOMES

1	Apply control structures, functions and packages in Problem Solving	L3
2	Analyze various String handling functions and data structures	L4
3	Model the object-oriented problem s with classes and objects.	L4
4	Solve the problems by using Inheritance and polymorphism.	L3
5	Illustrate programs on Exception-Handling and various packages	L3

C. Visualization

Python Tutor's code visualizer proves invaluable for the analysis and understanding of Python code through visualization. Users can step through code execution line by line, a critical capability for both debugging and educational purposes. By visualizing variables and their values at each step, Python Tutor provides clarity on how data flows through a program, aiding in the identification of logic errors and improvement of overall code comprehension. Furthermore, Python Tutor supports multiple Python versions and various languages such as Java, JavaScript, TypeScript, Ruby, C, and C++, enhancing its versatility for illustrating programming concepts and debugging complex algorithms across different environments. Whether educators leverage it to enhance student learning or developers use it to streamline coding processes, Python Tutor simplifies the understanding of code structure and behaviour, thereby boosting productivity in Python programming tasks.

Benefits

1. Enhanced Understanding: Python Tutor's visualizations offer a clear, step-by-step breakdown of code execution, helping users grasp intricate program flows and logic.
2. Efficient Debugging: By visualizing variables and their values dynamically, Python Tutor accelerates the debugging process, swiftly identifying and rectifying logic errors.
3. Educational Tool: It serves as a valuable aid for educators, facilitating the illustration of programming concepts with real-time code-visualization.
4. Cross-Language Support:Python Tutor's versatility extends to multiple programming languages beyond Python, broadening its utility for developers working across diverse tech stacks.

D. Jigsaw Puzzle:

A jigsaw is a puzzle game which involves solved pieces that are irregular in shape in a way that they fit together to form a whole picture. Each part usually consists of a fragment of an image, and the goal of the game is to collect parts and recreate the picture. Solving jigsaw puzzles programmatically in Python can be done through the use of NumPy for image processing and array operations. For example, NumPy enables developers to load, crop, resize, or manipulate the puzzle pieces in array form which is crucial in tasks such as segmenting the pieces from the images and basic operations such as converting the pieces to grayscale, edge detection, etc., and some feature extraction such as colour histogram and texture. The NUMPY operations assist to compare and position the puzzle pieces in relation to their attributes and relationships so that the efficiency of the puzzle-solving algorithms will be improved. By combining NumPy with libraries such as OpenCV and matplotlib, it becomes possible to create applications that can be used to solve puzzles both interactively and using programming language, making Python a useful tool for both enthusiasts and developers.

Benefits of Jigsaw puzzle

1. Some benefits that can be seen while using NumPy for jigsaw puzzles in Python are as

follows. Involves performing operations such as resizing and applying various filters to the puzzle images. Other features that make up an image or video, such as color histograms and edge maps, are also derived from NumPy to help in the process of identifying and matching pieces based on their appearance. Its optimized operations enhance the efficiency of several algorithms that are used to search for, match, and align pieces of the puzzles. NumPy is well compatible with image processing libraries like the OpenCV for operations such as edges to work out. In summary, NumPy improves interactions and the creation of engaging and learning-based jigsaw puzzle apps in Python, which helps in developing competencies such as spatial reasoning and problem-solving.

TABLE 2
COURSE OUTCOMES

1	Understand the basics and function of Python Programming Language.
2	Understand the string operation and sequences used in Python Programming Languages.
3	Understand the data structures used in Python Programming Languages.
4	Know the classes and objects in Python Programming Language.
5	Use the reusability concepts in Python Programming Language.

Fig 1. Group

Activity

Fig 2: CSE students presenting their concept

IV. RESULTS

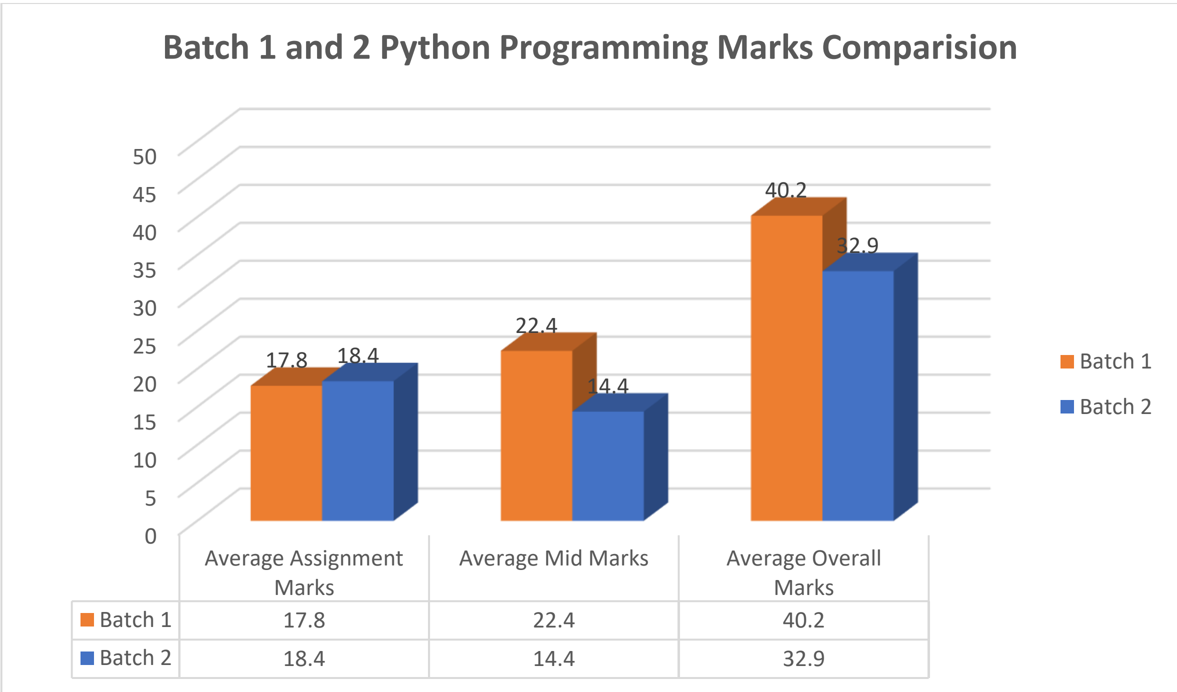

Fig 3. Comparing Average Marks of Batch 1 and Batch 2 Students

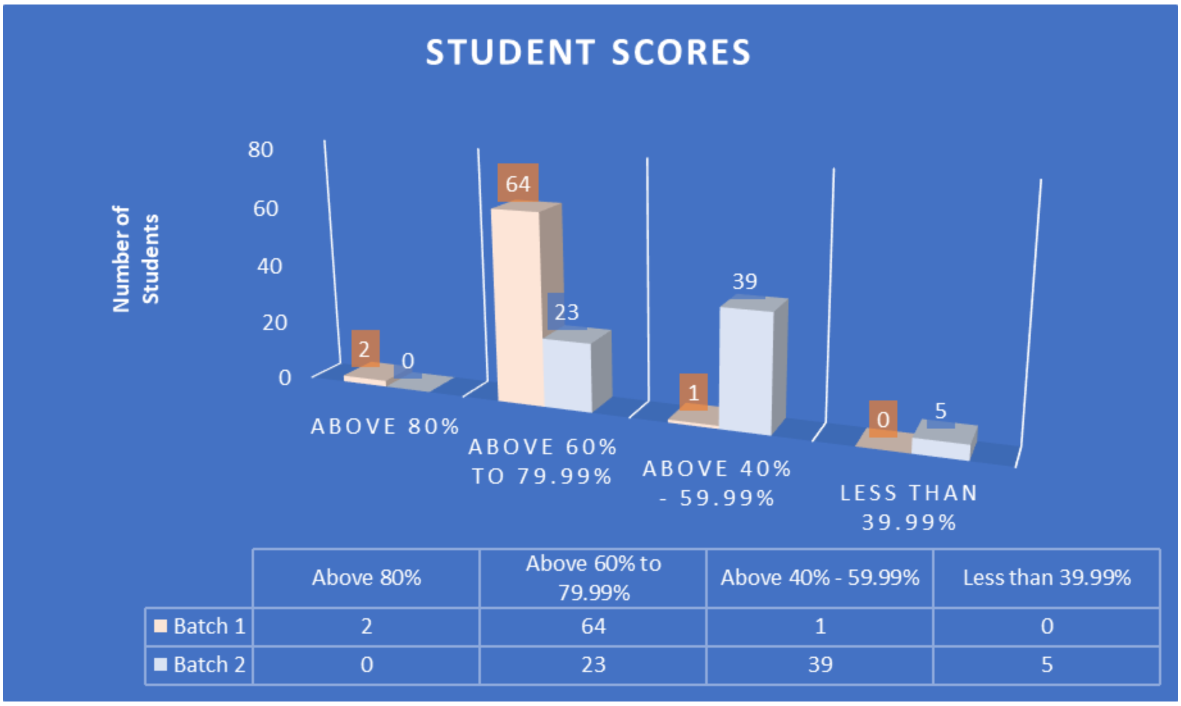

Fig 4. Student Scores

E and Batch 2 students Assignment Marks, Mid Marks, and Overall Marks are plotted for better understanding. In assignments, both sections have similar marks: Batch 1 got 17.8, and Batch 2 got 18.4. For mid exams, Batch 1 did much better with an average of 22.4, while Batch 2 got 14.4. Overall, Batch 1 has higher average marks at 40.2, compared to Batch 2's 32.9. This means that while both sections perform and Less than 39.99%. In the Above 80% range, Batch 1 has 2 students, while Batch 2 has none. For the 60% to 79.99% range, Batch 1 has many more students (64) compared to Batch 2 (23). In the 40% to 59.99% range, Batch 2 has many students (39), but Batch 1 has only 1. Lastly, in the Less than 39.99% range, Batch 2 has 5 students, and Batch 1 has none. This shows that most students in Batch 1 score between 60% and 79.99%, while Batch 2 students have scores spread more widely, with many scoring between 40% and 59.99%. Using TLP techniques for teaching Batch 1 students has improved their performance compared to Batch 2 students.

V CONCLUSION

The paper demonstrated the effectiveness of integrating diverse teaching-learning processes to enhance student engagement and comprehension. Methods like Think-Pair-Share, Quality Circles, and Python Tutor visualizations

similarly in assignments, Batch 1 does better in mid-term exams and overall, probably because of the TLP techniques used, which helped improve Batch 1's performance.

The chart compares student scores of two sections, E and G, in four ranges: Above 80%, 60% to 79.99%, 40% to 59.99%,

significantly improved student performance and participation. The comparative analysis showed that Batch 1, which utilized these interactive techniques, outperformed Batch 2 in mid-term and overall scores. The findings suggest that incorporating interactive and technology-driven pedagogies can foster deeper understanding and retention, making the learning process more engaging and effective for students. Future research should explore the long-term impacts of these methods on students' academic and professional success. Engagement and comprehension. Methods like Think-Pair-Share, Quality Circles, and Python Tutor visualizations significantly improved student performance and participation. The comparative analysis showed that Batch 1, which utilized these interactive techniques, outperformed Batch 2 in mid-term and overall scores.

REFERENCES

Freeman, S., Eddy, S.llearning increases student performance in science, engineering engineering, and mathematics. PNAS Proceedings of the National Academy of Sciences of the United States of America, 111(23), 8410–8415.

Kordaki, Maria & Siempos, Haris. (2010). The Jigsaw Collaborative Method within the Online Computer Science Classroom.. CSEDU 2010 - 2nd

International Conference on Computer Supported Education, Proceedings. 2. 65-72.

Jeskova, Zuzana & Lukac, Stanislav & Snajder, Lubomir & Guniš, Ján & Klein, Daniel & Kireš, Marian. (2022). Active Learning in STEM Education with Regard to the Development of Inquiry Skills. Education Sciences. 12. 686. 10.3390/educsci12100686.

Abirami A.M , Pudumalar S ThiruchadaiPandeeswari S, "Active Learning Strategies and Blended Learning Approach for Teaching Under Graduate Software Engineering Course", Journal of Engineering Education Transformations , Volume 35 , No. 1 , July 2021 , ISSN 2349-2473, eISSN 2394-1707.

Preeti S. Joshi, Visualization and Crossword: In-class Exercise---An active learning Strategy, Journal of Engineering Education Transformations , Volume 35 , No. 2 , October 2021 , ISSN 2349-2473, eISSN 2394-1707.

Sandeep U. Mane , Satyajit R. Patil , Sushma S. Kulkarni, Online Teaching-Learning xperience During E COVID-19 Pandemic – A Case Report, Journal of Engineering Education Transformations , Volume 35 , No. 2 , October 2021 , ISSN 2349-2473, eISSN 2394-1707.

Emmanuel James P. Pattaguan, Do Engineering Educators Read?: Appraising a University Reading Program, Journal of Engineering Education Transformations , Volume 35 , No. 2 , October 2021 , ISSN 2349-2473, eISSN 2394-1707.

T. Sujithra , N.M.Masoodhu Banu, Gaming Pedagogy For Effective Learning In Engineering Colleges, Journal of Engineering Education Transformations , Volume 35 , No. 2 , October 2021 , ISSN 2349-2473, eISSN 2394-1707.

Julius Fusic S, Online assignment approach in mechatronics system design course using Google classroom, Journal of Engineering Education Transformations , Volume 35 , No. 2 , October 2021 , ISSN 2349-2473, eISSN 2394-1707.

Muhammad Nurtanto , Herminarto Sofyan , Pardjono Pardjono, E-Learning Based AutoCAD 3D Interactive Multimedia on Vocational Education (VE) Learning. Journal of Engineering Education Transformations , Volume 34 , No. , 2021, ISSN 2349-2473, eISSN 2394-1707 .