RAMCO INSTITUTE OF TECHNOLOGY
Approved by AICTE, New Delhi & Affiliated to Anna University
Accredited by NAAC & An ISO 9001:2015 Certified Institution
NBA Accredited UG Programs: CSE, EEE, ECE and MECH

Department of Computer Science and Engineering
Academic Year 2024 – 2025 (Odd Semester)

Degree, Semester & Branch: III Semester B.E EEE
Course Code & Title: CS3353 & C Programming and Data Structures
Name of the Faculty member (s): Mrs.M.Dhivya AP/CSE

Innovative Practice Description

1. **Unit / Topic: Unit III / Application of Stack**

- **Course Outcome:** CO3
- **Topic Learning Outcome:** TL09
- **Activity Chosen:** Think-Pair-Share

Justification:

Application of Stack is such an interesting and important topic, by thinking individually first, students can formulate their own ideas about how stacks are used in various programming contexts. Pairing up encourages them to challenge and refine their thoughts. Students may have different experiences or knowledge regarding stack applications. Sharing these insights broadens everyone's understanding and highlights the versatility of stacks in various fields like data structures, algorithms, and even real-world applications.

Time Allotted for the Activity: 20 Minutes

Procedure of the Implementation:

- Think-pair-share is a technique that encourages and allows for individual thinking, collaboration, and presentation in the same activity.
- T (Think): Teacher asks a specific question about the topic. Begin by posing open-ended questions about the topic or text to encourage deeper thinking. Give students time to pause, reflect, and recall what they already know. This moment of reflection helps them form a personal response to the question. [Takes 1-5 Minutes].
- P (Pair): Each student should be paired with another student or place them in small groups. This collaboration helps students refine their thoughts through discussion, giving them the opportunity to clarify their understanding by talking with peers . [Takes 2-5 Minutes].
- S (Share): Students share their ideas with their partner. Afterward, teachers can facilitate a whole-class discussion, expanding on individual contributions and deepening the overall class engagement. [Takes 05-10 minutes].

Details of the Implementation

Think: 1-5 Minutes

- Faculty asks the following questions:
    - What are some real-world applications of stacks?
    - How do stacks function in programming contexts?
    - Can you think of an example from personal experience where a stack might be useful?
- Students think about these questions, ideas, and examples.

Pair: 2-5 Minutes

- Students form pairs and share their individual thoughts.

  Faculty encourage them to discuss:
  - Differences in their examples.
  - Any applications they hadn't considered.
  - How stacks could be implemented in the examples they discussed.
- Each pair summarizes their discussion points on paper.

Share: 5-10 Minutes

- Pairs join with other pairs (forming small groups) and share their findings.
- From that each group one student present about what they discussed about these following questions.

What are some real-world applications of stacks?

- Web browsers use stacks to manage page navigation. When a user visits a new page, it is pushed onto the stack. Clicking the back button pops the current page, allowing the user to return to the previous one.
- Stacks keep track of active function calls in programming languages. Each time a function is called, its execution context (local variables, return address) is pushed onto the stack. When the function completes, its context is popped off the stack.
- Stacks are used in calculators and programming languages to evaluate expressions, especially in converting infix expressions (like A+B) to postfix (like AB+). This is essential for parsing and computation.

How Stacks Function in Programming Contexts

- Stacks operate on the Last In, First Out (LIFO) principle, where the most recently added item is the first to be removed.
- Common operations include push (add an element), pop (remove the top element), peek (view the top element without removing it), and checking if the stack is empty.
- Stacks can be implemented using arrays or linked lists. In recursive function calls, the call stack maintains the order of function execution, enabling proper return flow and memory management.

Can you think of an example from personal experience where a stack might be useful?

- When I visit different websites, each page I navigate to is stored in my browser's history. If I click on a link and then realize I want to go back to the previous page, I can use the back button. This action relies on a stack structure: the most recent page I visited is at the top of the stack, and when I click "back," that page is popped off the stack, taking me back to the last page I viewed. This stack-based approach allows me to easily navigate through my browsing history without losing track of where I've been. It's a great example of how stacks help manage state in real-time applications.
- In a personal context, consider using a text editor while writing an essay. When drafting, I often make edits—adding and deleting sentences or phrases. Each change I make could be pushed onto an undo stack. If I decide I don't like a recent change, I can simply hit "undo," which pops the last action off the stack and restores the previous version. This functionality helps me experiment with different ideas without the fear of permanently losing my work.
- Finally, teacher and students discussed the answers of all questions at the end of the activity.

CO – PO / PSO mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO9	PO10	PO12	PSO1	PSO2
CO3	3	3	2	2	1	1	2	1	2	2

(1 – Low 2 – Moderate 3 – High)

Justification for correlation	
PO1	Students will be able to apply linear data structure concepts to identify solution to complex engineering problems
PO2	Students will be able to Identify, and analyze engineering problems with queue and stack data structures concept
PO3	Students will be able to design solutions for complex engineering problems using different data structures operations
PO4	Students will be able to provide complex problem solutions with relevant data structure concepts and operations
PO5	Students will be able to provide complex problem solutions with relevant data structure concepts and operations
PO9	Students will be able to use Visualgo.net tool for performing various operations of linear data structures linked list, stack, and queue.
PO10	Students will be able to write program individually to perform different operations of linear data structures.
PO12	Student will become aware of the need for lifelong learning and the continued upgrading of technical knowledge using the concept of linear data structures
PSO1	Students will be able to apply linear data structures concept to solve various application in modern software packeges
PSO2	Students will be able to specify the electronic system with linear data structure

- **Images / Screenshot of the practice:**

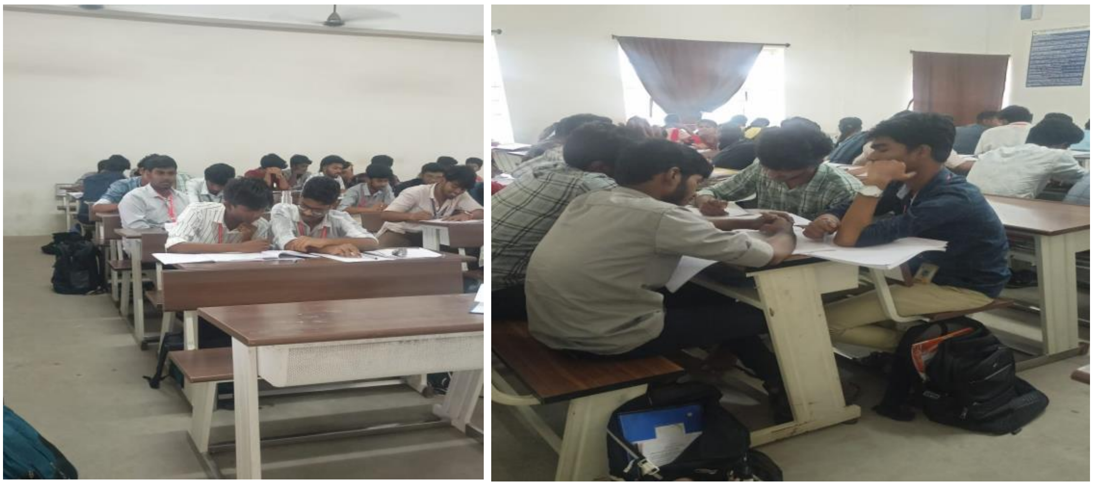

Figure 1 – Students Think and Share their Thoughts and Topic Presentation

- **Reflective Critique:**

Feedback of practice from students and other stakeholders:

- ❖ Some students reported feeling more confident in their ability to articulate their thoughts after discussing with a partner. This peer interaction made them more comfortable sharing in larger groups.
- ❖ Students appreciated discovering a variety of real-world applications of stacks they hadn't considered before, such as their use in web browsers and text editors.
- ❖ Bright students enjoyed with peer learning.

Benefit of the practice:

- Think-pair-share is a simple technique that enhances students' critical thinking skills, improves listening and reading comprehension, and helps with collaboration and presentation skills.
- Students can learn from each other during the "pair" phase by exchanging ideas and perspectives.
- students who are outspoken will benefit from first listening to others before sharing their own opinion.
- Gives every student a chance to participate, rather than relying on only a few vocal individuals.

Challenges faced in implementation:

- In some pairs, one student may dominate the conversation, while the other may not contribute as much. This can lead to imbalances in learning and engagement.
- Some students may feel shy or hesitant to share their ideas in front of peers, which can limit the effectiveness of the activity and hinder their learning experience.
- Without specific prompts or examples, students may struggle to think of relevant applications of stacks, leading to less productive discussions.
- Off-topic discussions may arise if students are not closely monitored or engaged.

References:

1. adlit.org, "Adolescent Literacy", 2019. [Online]. Available: http://www.adlit.org/strategies/23277/ [Accessed: 23.01.2020]
2. Reading Books - https://www.readingrockets.org/strategies/think-pair-share
3. prodigygame.com, "active-learning-strategies", 2019. [Online]. Available: https://www.prodigygame.com/blog/active-learning-strategies-examples/.
4. Reading Books- Centre for Teaching and Learning https://www.kent.edu/ctl/think-pair-share
5. https://www.edutopia.org/article/think-pair-share-variations-16-ways-up-your-game