<https://www.interviewcake.com/concept/java/stack>

1. [Stack Implementation](https://www.techiedelight.com/stack-implementation/)
2. [Stack Implementation using Linked List](https://www.techiedelight.com/stack-implementation-using-linked-list/)
3. [Check if given expression is balanced expression or not](https://www.techiedelight.com/check-given-expression-balanced-expression-not/)
4. [Find duplicate parenthesis in an expression](https://www.techiedelight.com/find-duplicate-parenthesis-expression/)
5. [Evaluate given postfix expression](https://www.techiedelight.com/evaluate-given-postfix-expression/)
6. [Decode the given sequence to construct minimum number without repeated digits](https://www.techiedelight.com/decode-the-given-sequence-construct-minimum-number-without-repeated-digits/)
7. [Design a stack which returns minimum element in constant time](https://www.techiedelight.com/design-stack-which-returns-minimum-element-constant-time/)
8. [Design a stack which returns minimum element without using auxiliary stack](https://www.techiedelight.com/design-a-stack-which-returns-minimum-element-without-using-auxiliary-stack/)
9. [Reverse a string without using recursion](https://www.techiedelight.com/reverse-string-without-using-recursion/)
10. [Reverse a string using stack data structure](https://www.techiedelight.com/reverse-a-string-using-stack-data-structure/)
11. [Inorder Tree Traversal | Iterative & Recursive](https://www.techiedelight.com/inorder-tree-traversal-iterative-recursive/)
12. [Preorder Tree Traversal | Iterative & Recursive](https://www.techiedelight.com/preorder-tree-traversal-iterative-recursive/)
13. [Postorder Tree Traversal | Iterative & Recursive](https://www.techiedelight.com/postorder-tree-traversal-iterative-recursive/)
14. [Find ancestors of given node in a Binary Tree](https://www.techiedelight.com/find-ancestors-of-given-node-binary-tree/)
15. [Check if two given binary trees are identical or not | Iterative & Recursive](https://www.techiedelight.com/check-if-two-binary-trees-are-identical-not-iterative-recursive/)
16. [Reverse Level Order Traversal of Binary Tree](https://www.techiedelight.com/reverse-level-order-traversal-binary-tree/)
17. [Reverse given text without reversing the individual words](https://www.techiedelight.com/reverse-text-without-reversing-individual-words/)
18. [Find all binary strings that can be formed from given wildcard pattern](https://www.techiedelight.com/find-binary-strings-can-formed-given-wildcard-pattern/)
19. [Iterative Implementation of Quicksort](https://www.techiedelight.com/iterative-implementation-of-quicksort/)
20. [Depth First Search (DFS) | Iterative & Recursive Implementation](https://www.techiedelight.com/depth-first-search/)
21. [Invert given Binary Tree | Recursive and Iterative solution](https://www.techiedelight.com/invert-binary-tree-recursive-iterative/)
22. [Print leaf to root path for every leaf node in a binary tree](https://www.techiedelight.com/print-leaf-to-root-path-binary-tree/)
23. [Longest Increasing Subsequence](https://www.techiedelight.com/longest-increasing-subsequence/)

<https://www.interviewbit.com/courses/programming/topics/stacks-and-queues/>