



Problem Solving With C - UE24CS151B

Stack

Prof. Sindhu R Pai

PSWC Theory Anchor, Feb-May, 2025

Department of Computer Science and Engineering

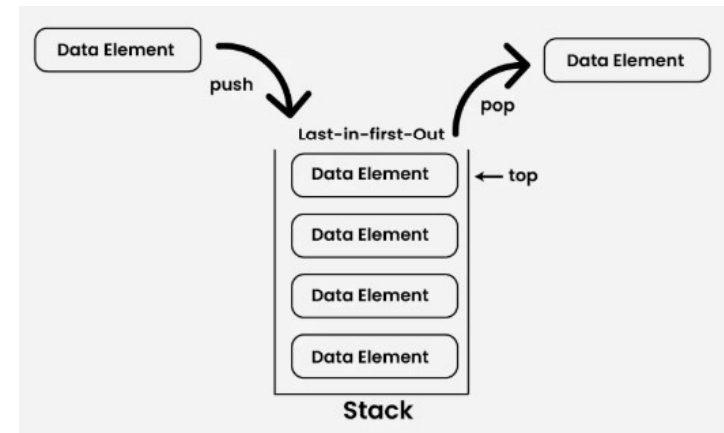
PROBLEM SOLVING WITH C

Stack



What is a stack?

- Data structure that stores data using the **Last In First Out (LIFO)** principle
- Imagine a stack of books. The last book you put on top of the stack (LI) is the first one you take off the stack (FO)
- Linear data structure: Stores data linearly or sequentially



PROBLEM SOLVING WITH C

Stack

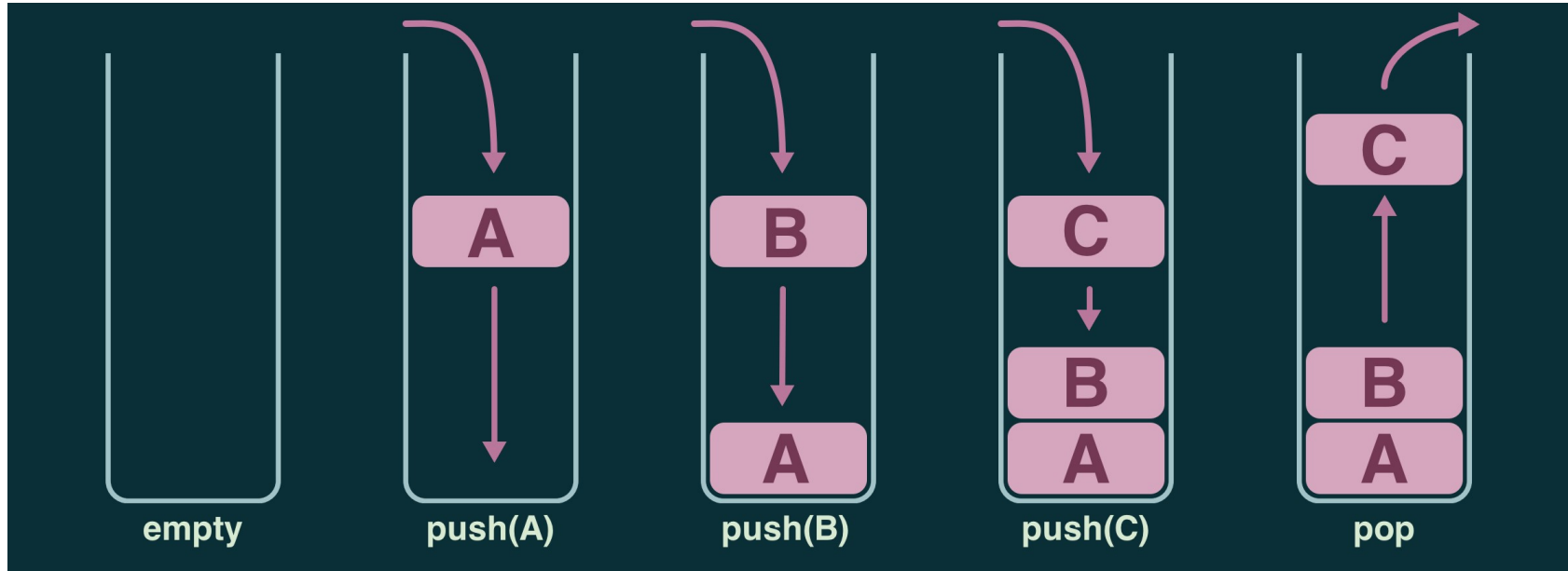


Key Terms and Operations

- **Top:** The end of the stack where modifications are made
- **Push:** Adds an element to the top of the stack.
- **Pop:** Removes the element from the top of the stack.
- **Peek:** Returns the top element without removing it.
- **IsEmpty:** Checks if the stack is empty.
- **IsFull:** Checks if the stack is full (for fixed-size stacks).

PROBLEM SOLVING WITH C

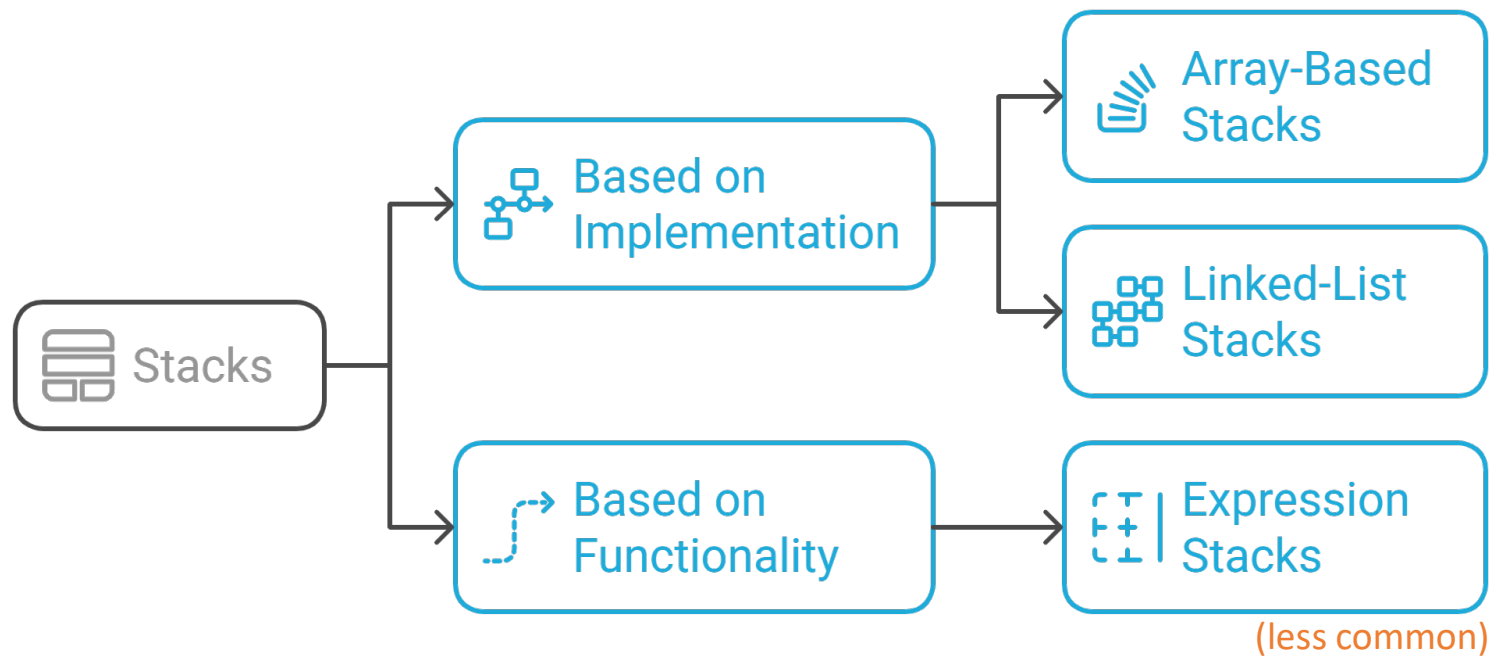
Stack



PROBLEM SOLVING WITH C

Stack

Types of Stacks



PROBLEM SOLVING WITH C

Stack



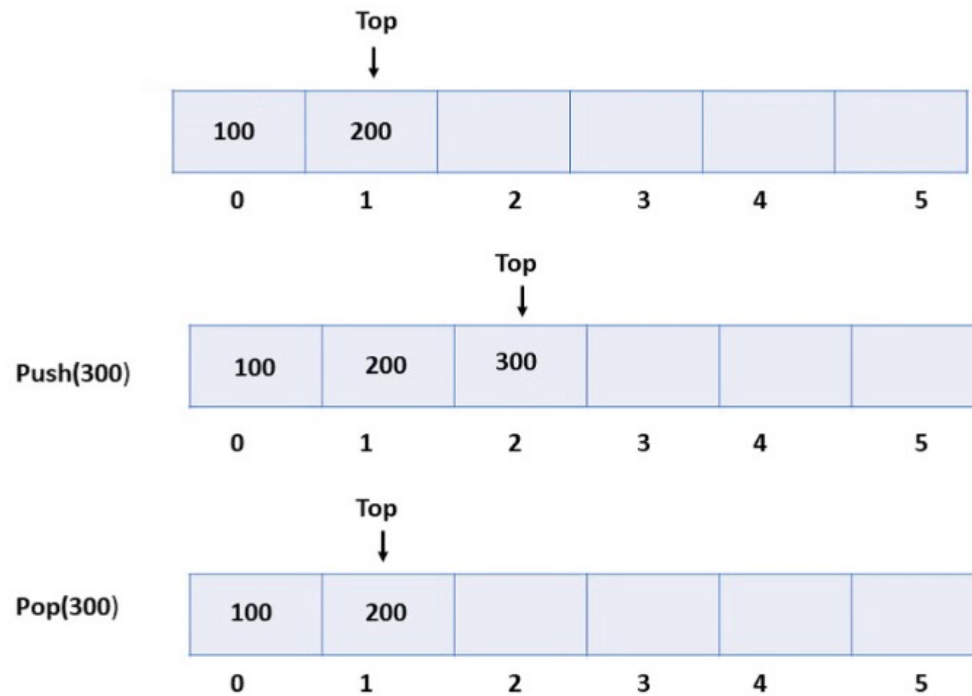
Array Based Stack

- **Data Structure:** Array of fixed size
- **Variables:** Top (index of the top element, usually initialized to -1)
- **Operations:**
 - Push: Increment top, add element at array[top]
 - Check for overflow (IsFull condition)
 - Pop: Return element at array[top], decrement top
 - Check for underflow (IsEmpty condition)
 - Peek: Return element at array[top] without modifying top

PROBLEM SOLVING WITH C

Stack

Array Based Stack



PROBLEM SOLVING WITH C

Stack



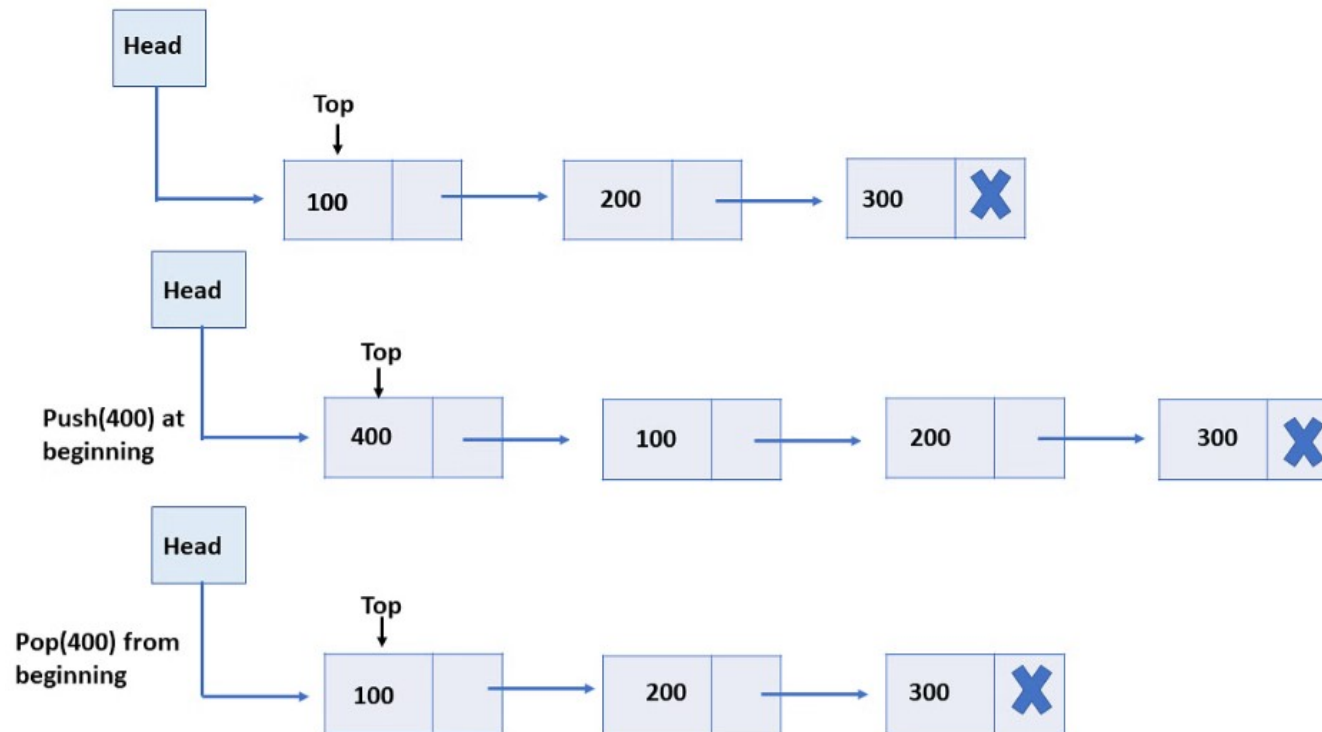
Linked List Stack

- **Data Structure:** Singly linked list
- **Variables:** Top (pointer to the top node)
- **Operations:**
 - Push: Create a new node, set its next to top, update top to the new node.
 - Pop: Store the top node's data, update top to top.next, free the old top node.
 - Peek: Return the data of the top node.
 - IsEmpty: Check if top is NULL/None.

PROBLEM SOLVING WITH C

Stack

Linked List Stack



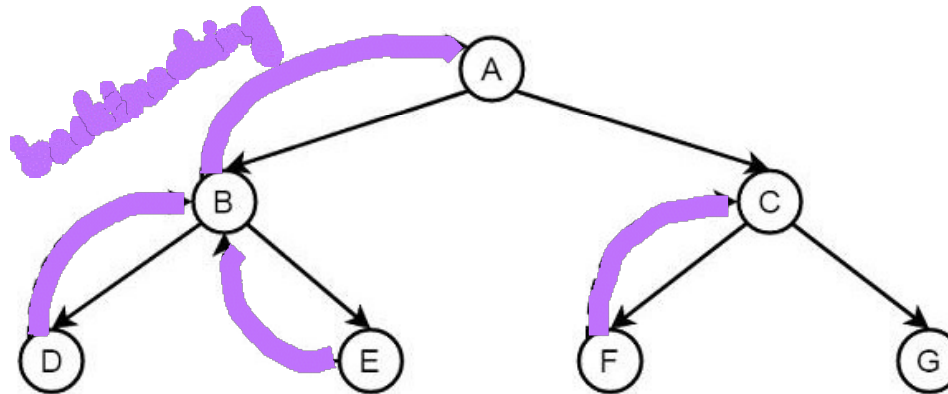
PROBLEM SOLVING WITH C

Stack



Applications

- **Backtracking** is implemented using stacks. This is a recursive algorithm that is used to solve optimization problems



- Stacks are also used to **evaluate expressions written in postfix/prefix notations**, and to convert infix to postfix/prefix (this will be covered in a later course).

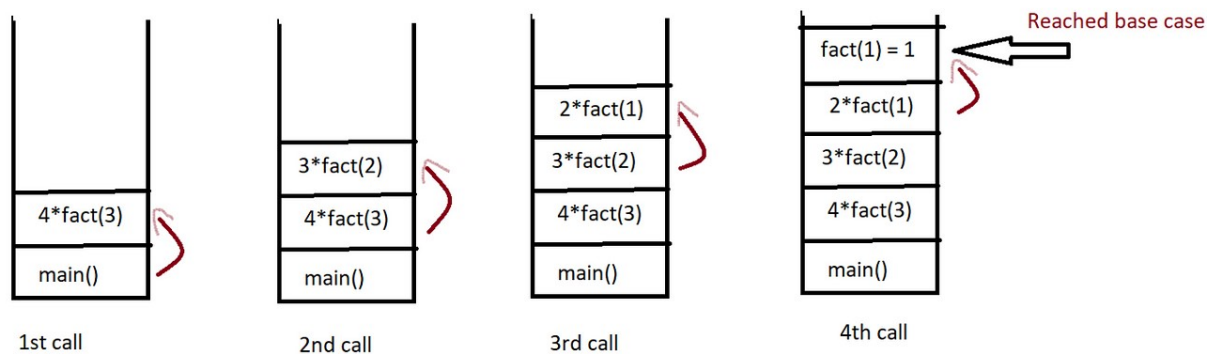
PROBLEM SOLVING WITH C

Stack



Applications

- **Function calls are kept track of** using stacks in the computer system. Every call results in a record being added to the stack, which is then popped once the function returns.
- The above naturally extends to the concept of **recursion** as well; this is also kept track of using stacks.



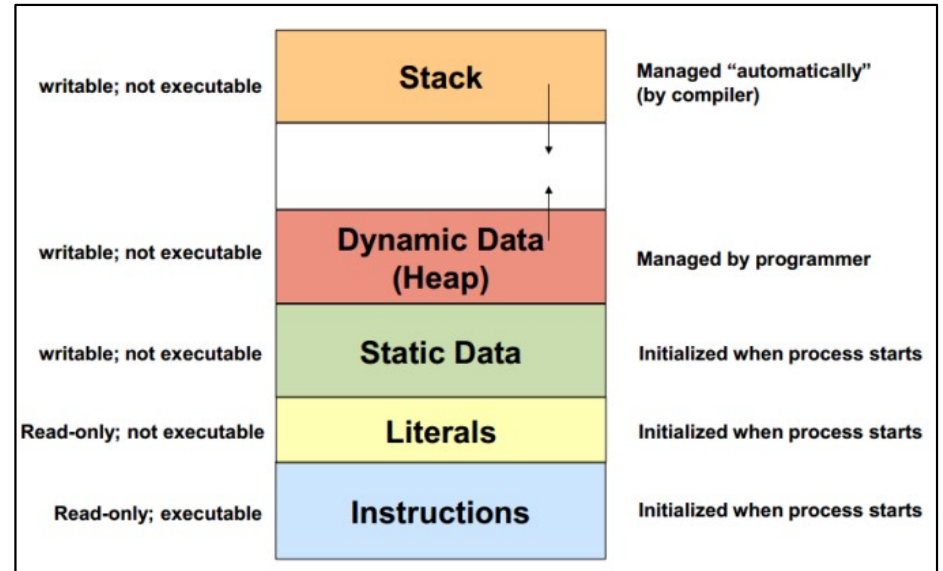
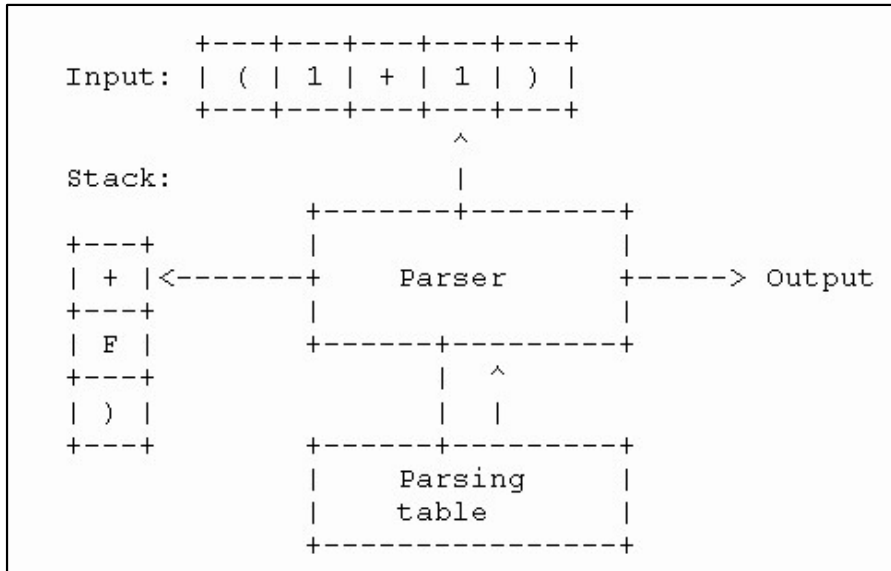
PROBLEM SOLVING WITH C

Stack



Applications

Stacks are also used for **syntax parsing** in compilers and **memory management** in operating systems(For further exploration)





THANK YOU

Department of Computer Science and Engineering

Dr. Shylaja S S, Director, CCBD & CDSAML, PESU
Prof. Sindhu R Pai - sindhurpai@pes.edu

Ack: Teaching Assistant – Advait Sanil Kumar