C++
Stacks

CO658 Data Structures & Algorithms

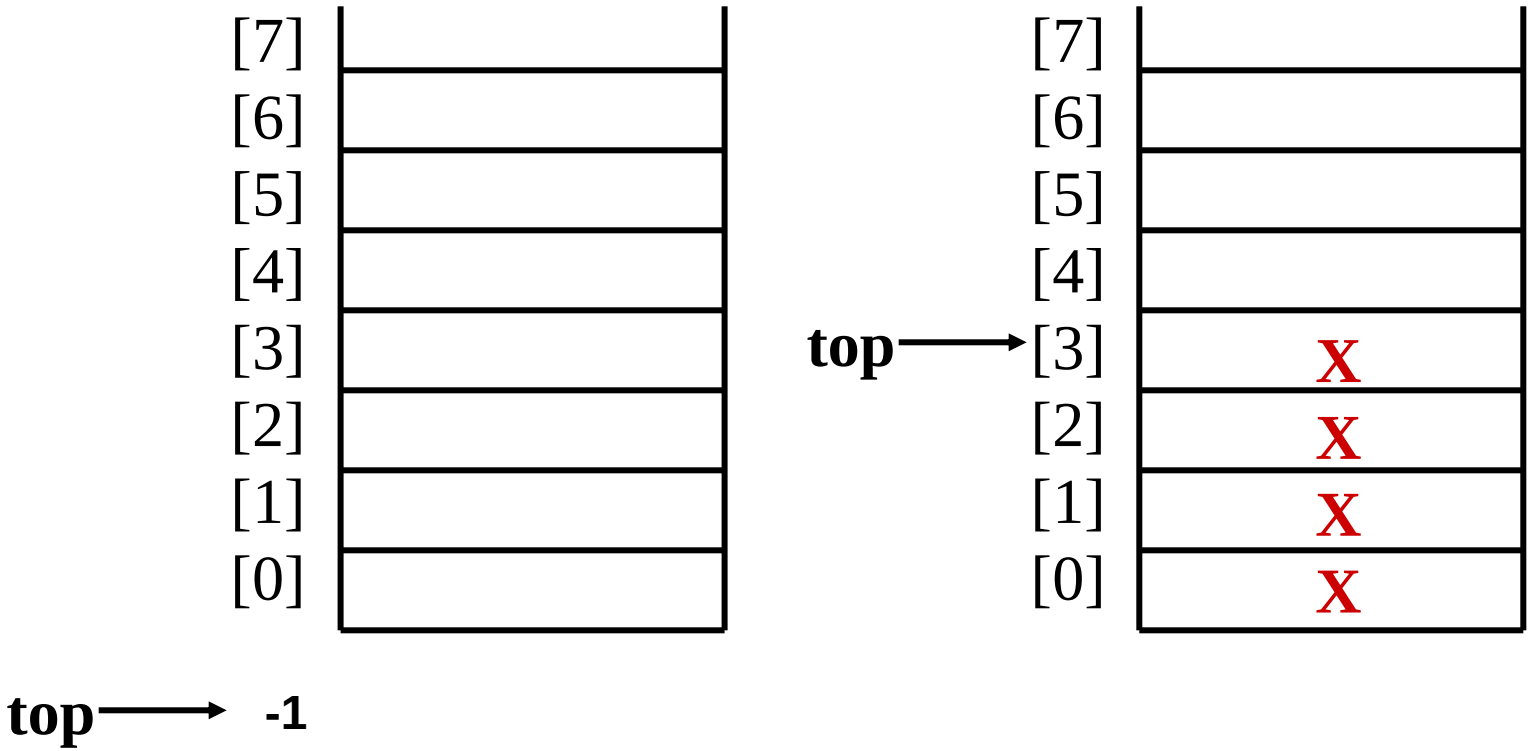
Topics

- Stacks

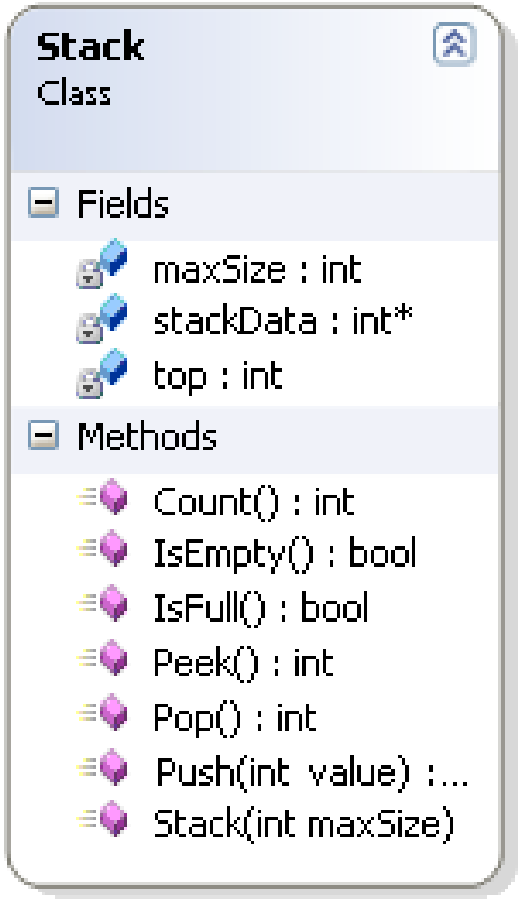
Stack

- Allows access to one data item, the last item inserted into the stack.
- Last In First Out LIFO, only the top data item is ever visible.
- If you remove this you have access to the next data item etc.
- The structure used to implement the stack is hidden, although arrays are frequently used.
- A Stack is often implemented as an array, because manipulation does not involve moving data items, other than the top
- Popping the Stack actually leaves the data in its position and merely changes the Stack Pointer

Stack as Array



Stack Design



Stack Operations

Operations defined informally

Initialise the stack

- **Push** - add an item on to the stack, if the stack is not full
- **Pop** – remove the top element from the stack, if the stack is not empty
- **Peek** - get the top element from the stack, without removing it, if the stack is not empty.
- **IsFull** Check if the stack is full
- **IsEmpty** Check if the stack is empty

Stack Rules

- The newly initialised stack is empty
- Immediately after pushing an item on to the stack, the stack is not empty
- Immediately after pushing an item on to the stack, that item is on the top of the stack
- Immediately after pushing and then popping, the stack is the same as it was immediately before the pushing

Summary

- Both stacks and lists are Abstract Data Types.
- The ADT publishes an interface. The implementation however is hidden.
- Selecting the correct data structure is critical to the performance of the program.