The Stack ADT

- Ordered collection/container, last-in-first-out (LIFO) (i.e., removal in reverse order of insertion)
- Functionality:
    - **initialize**
    - **push**: add item at top of stack
    - **pop**: remove item from top of stack
    - **top**: get (copy) of item on top of stack
    - management: **size**, **isEmpty**, etc.

Class Invariant

- Explicit statement of how the data structure is used to represent the ADT (e.g., how the member variables are used)
- Essential to correct implementation of class's functions
- **Each function depends on class invariant being valid when function called**

- Stack class implemented with **array** for *data*

  (1) No. of items in stack stored in member variable *used*
  (2) Items in stack stored in partially filled array called *data*
  (3) Bottom of stack at *data[0];* top of stack at *data[used – 1]*

- Stack class implemented with (pointer-based) **linked list** for *data*

  (1) Items in stack stored in a linked list, with top of stack at head node and bottom of stack at tail node.
  (2) *topPtr* is the head pointer of the linked list of items

Overflow and Underflow Errors

- ***Stack overflow***: attempting to push an entry onto a full stack
- ***Stack underflow***: attempting to pop an entry off of an empty stack

Standard Template Library (STL) Stack Class

- *#include <stack>*
- stack<int> myStack;