Stacks

What is a stack?

- Last-in first-out data structure (LIFO)
- New objects are placed on top
- Removal restricted to top object
- Examples?

Stack ADT Operations

- **push**(o): Insert o at top of stack
  - Input: Object; Output: None
- **pop**(): Remove top object; error if empty
  - Input: None; Output: Object removed
- **size**(): Return number of objects in stack
  - Input: None; Output: Integer
- **isEmpty**(): Return a boolean indicating stack empty
  - Input: None; Output: Boolean
- **top**(): Return top object without removing; error if empty
  - Input: None; Output: Object

Example

- push(5)
- push(3)
- pop()
- push(7)
- pop()
- top()
- pop()
- pop()
- isEmpty()

- push(9)
- push(7)
- push(3)
- push(5)
- size()
- pop()
- push(8)
- pop()
- pop()

Implementing a Stack

- Implementation involves writing one or more classes which provide functions to accomplish stack operations

Stack Interface

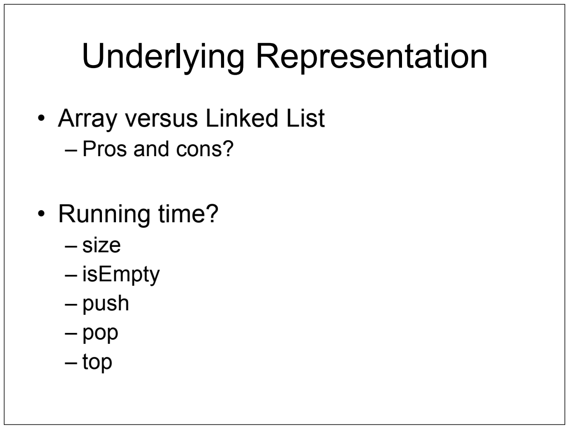
```
int size();

boolean isEmpty();

Object top() throws StackEmptyException;

void push(Object obj);

Object pop() throws StackEmptyException;
```



Underlying Representation

- Array versus Linked List
  - Pros and cons?
- Running time?
  - size
  - isEmpty
  - push
  - pop
  - top