CS 260 100P
Stacks

Revision Date: January 10, 2013

Printable Version

Topics to learn

Please read up on the following subjects:

- dynamic arrays
- circular arrays
- singly-linked lists
- tail pointers
- stacks
- queues
- public interfaces
- order notation (big Oh)

Tasks to perform

Implement the following classes:

- a stack class (based up a dynamic circular array class)
- a stack class (based up a singly linked-list class)

Each class should be implemented in a separate file (you will need to make a Node class and a List class, in addition to the stack classes, for this task). Provide a main function that tests the interfaces

of each class in a way that your Mentor can easily ascertain the correctness of your solution. Your main function should implement a mini-interpreter that responds to the following commands:

new TTT
: create a new stack of type TTT

push NNN
: push a value NNN onto the stack

pop
: pop a value from the stack and print the popped value

size
: display the current size of the stack

visualize
: print the entire stack

The mini-interpreter code can be written by your group and the code shared.

Descriptions of *Classes*, *Nodes*, *Lists*, and *Stacks* can be readily found in the literature. Your linked-list class must implement a tail pointer.

Other requirements

Be able to describe to your mentor the complexity of each operation in the public interface of your classes using order notation.

Note: most operations must be performed in constant time.