#### Stack ADT

#### section 4.13

## Abstract Data Type

An allowed set of values called the **domain** 

#### A specified set of **operations (API)**

Abstract:

- Users don't know how the data is stored
  - encapsulation
- Only access the data through the operations

### **Common Functionality**

insert(&dataStructure, key, data);

find(&dataStructure, key, &data);

insert(&dataStructure, "will4614", "chadd, 202 Strain, x3041");

## Goal

Build a useful data structure

Test the data structure

Hand that data structure off to a customer

- future you
- teammate
- paying customer
- your professor

# Types

- Basic
  - Array
  - Linked List
- Preserve structure
  - Stack
  - Queue
- Lookup by value
  - Tree
  - Hash Table

#### Stack

# The stack is a LIFO (Last-in First-out) data structure

#### Stacks

**Browser History** 

undo in applications

runtime stack

certain computations

## Operation

#### **Stack Operations**

#### Stack ADT Continued

#### Stack ADT Continued

# Testing your Data Structure

- Your customer will abuse your data structure
- Your data structure should never crash the customer's code

- code defensively

- Test each each function
  - test each functions *requires* statement
  - test boundary conditions (full/empty)
  - test bad input
  - test functions called in the wrong order

## Example

- Parentheses in algebraic expressions need to be balanced in order for the expression to be correct.
- Which of the following are valid expressions?

 How can a stack be used to test if an expression's parentheses are balanced?

## Assignment 1