Creating Classes in C++: Stack example

26 Jan 2009
CMPT166
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Quick note about pointers

- We can refer to attributes, methods of an object with the dot "." operator:
  - myVec.size()
- But remember that C++ always passes parameters as call-by-value
- So to pass an object, we need to pass a pointer
- To get to attributes/methods of an object from its pointer, use the arrow "->" operator:
  - myVectorPtr->size()
- Also, new always returns a pointer

Classes: declare vs. define

- (See Stack class in examples/ directory)
- Header file: Stack.h
  - Declare Stack class
  - public/private sections
  - Declare methods, including constructor (Stack()) and destructor (~Stack())
  - Defines helper class: Node
- Implementation: Stack.cpp
  - Define methods in Stack:: namespace
    - void Stack::push(void* dat)

Stack.h: header file

- Pre-processor include guards: #ifdef
- Helper class: Node: a node in a linked list
  - Attributes: data, next
    - void* data means payload can be anything
  - Methods: constructor, destructor
- Stack class: we implement using linked list
  - Attribute: Node* head (private)
  - Methods: push(), pop(), peek()
    - Also constructor, destructor
    - peek() returns top value without popping

Stack.cpp: define methods

- Define methods of both Node and Stack classes
- Must prefix names of methods with class names
  - each class has its own namespace
    - void Stack::push(void* dat)
- Define constructors, destructors
  - Con-/de-structors don't return anything
    - Stack::~Stack()
- Refer to attributes directly (no need for “self”)

Managing memory: who owns?

- Our linked-list is a dynamic data structure
  - Allocates memory on the heap
  - Must make sure to deallocate properly!
- The critical question is: who owns the object?
  - Who's responsible to deallocate its memory?
- Philosophy in Stack.cpp: calling program owns the data/payload, must deallocate it
  - So our destructors do not do any dealloc
  - So ~Stack() assumes stack must be empty!

StackTest.cpp: testbed

- Stack.cpp doesn't have a main() function
- StackTest.cpp is our testbed program
  - Has a main() function, creates a Stack
  - Creates payloads; deallocs them, too
- using namespace std; line goes in testbed
  - Not in header files! (would defeat the purpose of namespaces)
    - #include “Stack.h”
    - #include <iostream>
    - using namespace std;