Exam 3 Review

CS 2308
Spring 2015

Jill Seaman

Exam 3

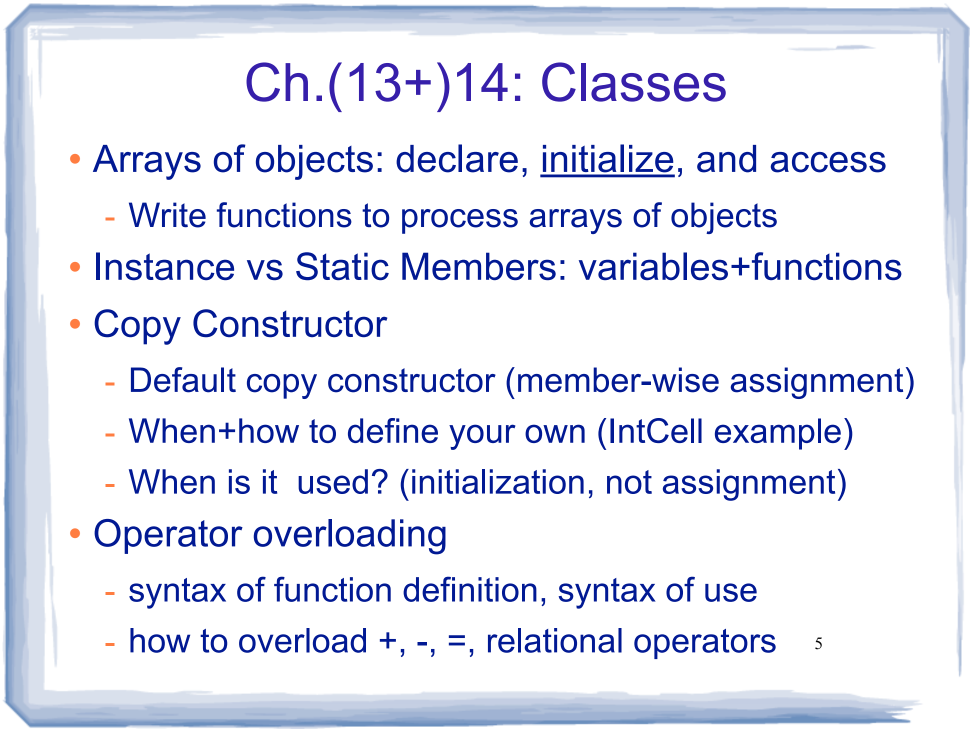
- Fri, April 24
- In class, closed book, closed notes, clean desk
- 15% of your final grade
- 50 minutes to complete it
- I recommend using a pencil (and eraser)
- All writing will be done on the test paper I will hand out. (I will bring extra paper).
- No calculators.

Exam Format

- 100 points total, 4 pages
  - Writing programs/classes/functions/code (lots of this)
  - Multiple choice
  - Fill-in-the-blank/short answer
  - Tracing code (what is the output)

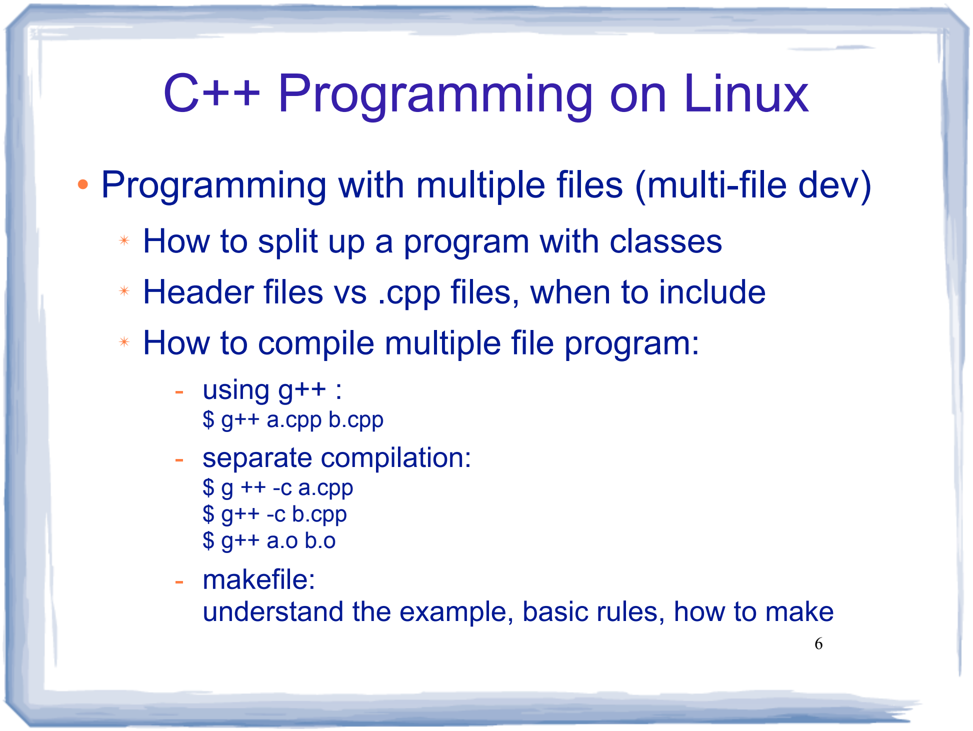
Which Lectures?

- Multi-file Development
- Chapter 14
- Pointers and this
- Chapter 17



Ch.(13+)14: Classes

- Arrays of objects: declare, initialize, and access
  - Write functions to process arrays of objects
- Instance vs Static Members: variables+functions
- Copy Constructor
  - Default copy constructor (member-wise assignment)
  - When+how to define your own (IntCell example)
  - When is it used? (initialization, not assignment)
- Operator overloading
  - syntax of function definition, syntax of use
  - how to overload +, -, =, relational operators

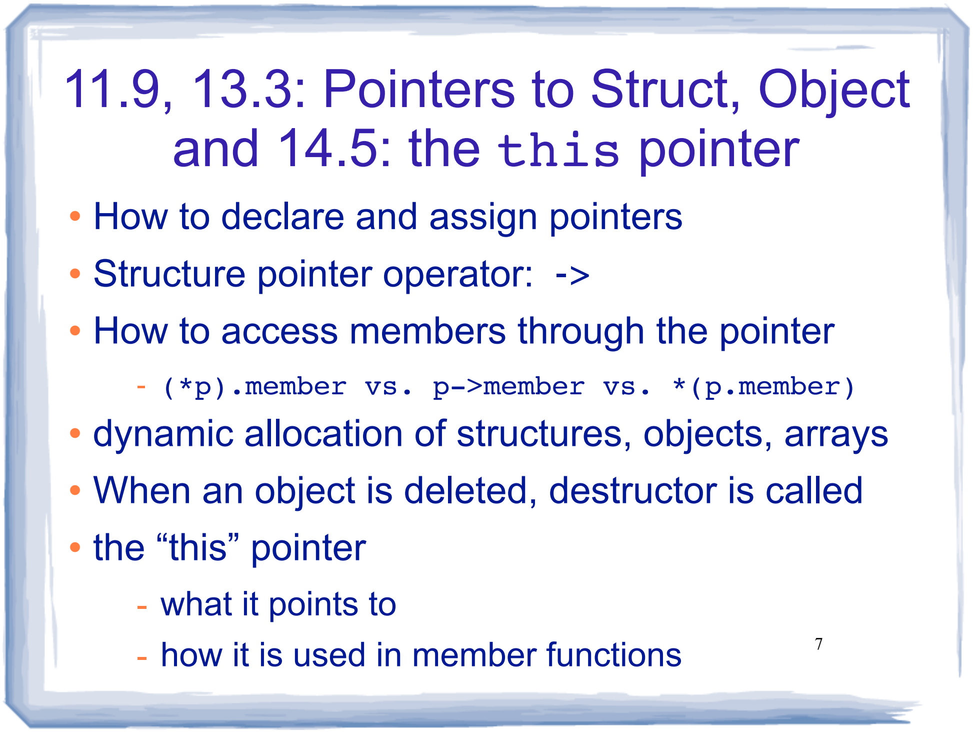


C++ Programming on Linux

- Programming with multiple files (multi-file dev)
  - How to split up a program with classes
  - Header files vs .cpp files, when to include
  - How to compile multiple file program:
    - using g++ :

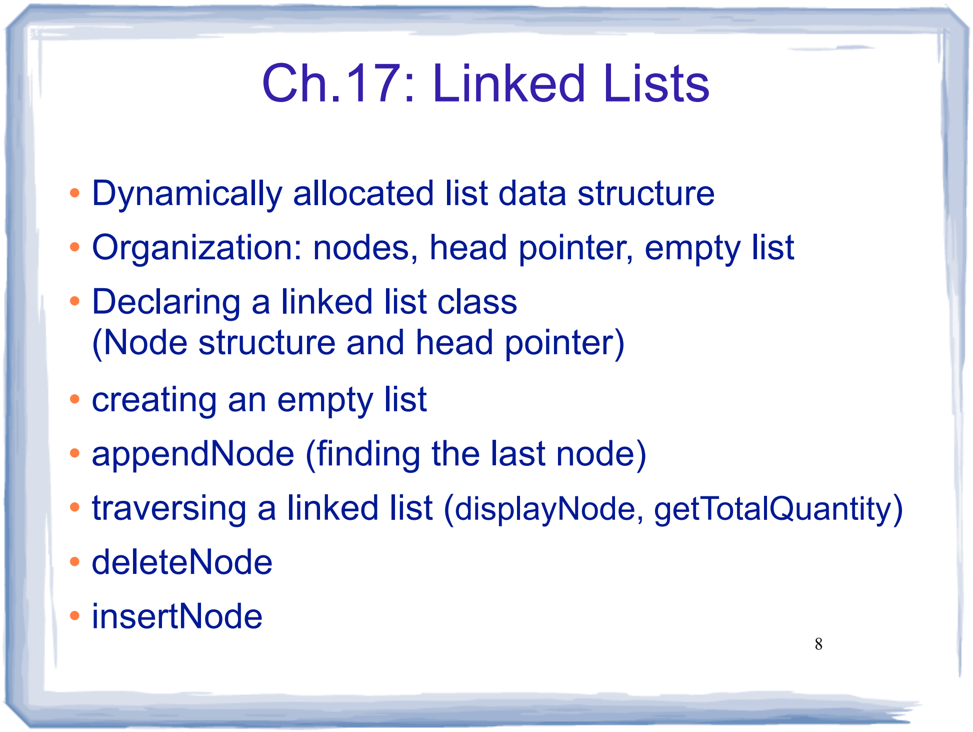
      ```
      $ g++ a.cpp b.cpp
      ```
    - separate compilation:

      ```
      $ g ++ -c a.cpp
      $ g++ -c b.cpp
      $ g++ a.o b.o
      ```
    - makefile:
      understand the example, basic rules, how to make



11.9, 13.3: Pointers to Struct, Object and 14.5: the `this` pointer

- How to declare and assign pointers
- Structure pointer operator: ->
- How to access members through the pointer
  - `(*p).member vs. p->member vs. *(p.member)`
- dynamic allocation of structures, objects, arrays
- When an object is deleted, destructor is called
- the “this” pointer
  - what it points to
  - how it is used in member functions



Ch.17: Linked Lists

- Dynamically allocated list data structure
- Organization: nodes, head pointer, empty list
- Declaring a linked list class
  (Node structure and head pointer)
- creating an empty list
- appendNode (finding the last node)
- traversing a linked list (displayNode, getTotalQuantity)
- deleteNode
- insertNode

Example Programming Problem

Consider a Rectangle class, which stores a rectangle using two floating point values: width and height.

It has 2 constructors (one default, one with 2 arguments), and accessors and mutators for the width and height. It also has an overloaded < operator (a rectangle is less than another if its area is less than the other's area).

The default rectangle has width and length equal to 1. The other constructor takes the initial width and height. The rectangle also has a function area() that calculates the area of the rectangle.

(a) Write the class declaration.

(b) Implement all the class functions.

Example Tracing Problem

A)What will the EXACT output of the following program be?

```
class IntCell {
   private:
      int *storedValue;
   public:
      IntCell (int initialValue)
      { storedValue = new int;
        *storedValue = initialValue; }
      ~IntCell()
      {  delete storedValue;  }
      int read () const
      {  return *storedValue; }
      void write (int x)
      {  *storedValue = x;  }
};

void main() {
   IntCell object1(10);
   IntCell object2 = object1;
   object2.write(4);
   cout << object1.read() << endl;
   cout << object2.read() << endl;
}
```

B) Define a copy constructor for the IntCell class so that two instances of IntCell will **not** point to the same storedValue.

Example Linked List Problems

A. Write the C++ code to declare a structure to represent a Node in a linked list that stores strings. Declare a pointer named head to (always) point to the first element of the list.

B. Declare a temporary pointer named p and make it point to the first element of the list.

C. Write C++ code to create a new Node, put "hello" in it, and add it to the list as the first element.

D. Assume the list now has 0 or more elements. Write a loop that will traverse the list and output each string from the list on a separate line.

E. Assume the list has at least one element. Write a loop to add a new Node containing "the end" to the list as the last element .

F. Assume the list has at least one element. Write a loop that will traverse the list until if finds the node containing "X" and then removes that node from the list (assume that "X" IS in the list).

How to Study

- Review the slides
  - understand all the concepts, quiz yourself
- Use the book to help understand the slides
  - there will be no questions over material (or code) that is in the book but not on the slides
- Review programming assignments (fix yours!)
  - get printouts of solutions in my office
- Try some exercises from the book
- Practice, practice, practice
- Get some sleep