First: Questions from Lab

• Tell me what material I need to review Friday
• All you should need for recitation that you don’t already have
• Today’s lecture is for Assignment 2, and next week’s labs/recitations is Destructors
• Tell me what material I need to review Friday

First: Questions from Lab
The Big 3 - Default Behavior

- Destructor - Call the destructors of all the object's class-type members.

- Assignment operator - Shallow copy, calling the copy assignment operators of the object's class-type members, and doing a plain assignment of all non-class type (e.g., int or pointer) data members.

- Copy constructor - Shallow copy, calling the copy constructors of the object's class-type members, and doing a plain assignment of all non-class type (e.g., int or pointer) data members.
Shallow Copy vs. Deep Copy

• Shallow: Copy the contents of member variables from one object to another.
  • Example: Shallow copy of Genre makes a pointer to the same movie array as in the Genre being copied.
  • Does not call new movie() constructor.
  • Default for assignment operator and copy constructor.

• Deep: Copy what each member variable is pointing to so that you get a separate but identical copy.
  • Example: Deep copy of Genre creates a new movie array with copies of the movies, and makes a pointer to the new array.
  • Calls new movie() constructor.
  • Specifically, does this because you program it to do so.

Shallow vs. Deep Copy
Copy Constructor

Copy constructor that has one parameter that is of the same type as the class type.

- Whenever an argument of the class type is "plugged in" for a call by value parameter.
- When a function returns a value of the class type.
- When a class object is being declared and initialized by another object of same type.
- Called automatically in three cases:
  - Allows for distinct copies, changes to one does not impact the other.
  - Has to be called by reference (normally const).

The class parameter that has one parameter that is of the same type as the class type.
Assignment Operator Overload

- Predefined assignment operator returns a reference
- Allows us to chain assignments together: `a = b = c`
- Need to make sure the assignment operator returns something of the same type as its left-hand side

Overloading assignment operator

- Must be a member of the class
- Overloads the predefined assignment operator
Destructor

• A delete operator is called
• A block containing local variables ends
• When the program ends
• When the function ends

• Called when the object goes out of scope
• ~Class_Name(); //no return type, no parameters, only one allowed
• Will not handle dynamic memory
• Will be automatically created if one is not supplied
• Deletes the object
Memory Leaks & Default Destructors

• Calling default destructor on genres()

  • Deletes ptr to movies[0], but not movie in movies[i]
  • How to delete movies[i]?

  • Can’t, needed to define Destructor

  • Calling default destructor on genres()
Note for Assignment 2:

We will cover "real" search/sorting algorithms in the final weeks of class. Searching in menu class really means iterating on an array and returning a new array with only the options we wanted.