Inheritance

Review
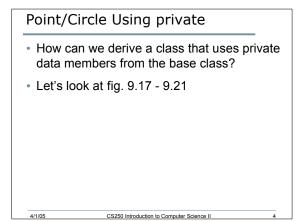
- How is inheritance performed in C++?
- Last time we created a Point class and a Circle class that inherited from Point
  - Write the class interface for Point
  - Write the class interface for Circle
  - Write the definition for the Circle constructor

Point/Circle Using private

- Using protected produced a slight increase in performance because we don't have the overhead of get and set function calls
- However, there are two problems
  - Derived classes can set illegal values to the base class' protected member
  - Derived classes member functions are more likely to be written based on the base class data members. Why is this a problem?

Point/Circle Using private

- How can we derive a class that uses private data members from the base class?
- Let's look at fig. 9.17 - 9.21

Problem

- There are two files in the CS250 pub folder:
  - `Time1.h`
  - `Time1.cpp`
    - a) Remove the method printUniversal from the class.
    - b) Replace the printStandard method from the class replacing the method with an overloaded stream-insertion operator.
    - c) Create a derived class called ZonedTime that inherits the behaviors and data of the base class time1.h. Add to this class the ability to specify the time based on one of the four zones: Pacific Standard Time, Mountain Standard Time, Central Standard Time, Eastern Standard Time.
    - d) Add a method to ZonedTime that outputs time including the time zone.
    - e) Overload the == operator such that two objects of ZonedTime can be compared to see if the times are equal or not. Remember, 5pm EST is the same as 2pm PST.

Summary

- We discussed inheritance
- We covered:
  - Pages 610 - 630