

---

## Inheritance and Polymorphism

4/4/05

CS250 Introduction to Computer Science II

1

---

## Three-Level Inheritance

- How would we create class **Cylinder** that inherits from class **Circle**?
- In what order are the constructors called when we create an object of type **Cylinder**?
- In what order are the destructors called?

4/4/05

CS250 Introduction to Computer Science II

2

---

## Inheritance

- This ends our discussion of inheritance
- We have completed chapter 9

4/4/05

CS250 Introduction to Computer Science II

3

---

## Polymorphism

- Program in the general, rather than program in the specific
- Two underlying technologies
  - Virtual functions
  - Dynamic binding

4/4/05

CS250 Introduction to Computer Science II

4

---

## Polymorphism in Action

- The example we will use will utilize the point/circle classes we created earlier
- For the print member function in class Circle we had the statement
  - `Point::print();`
- Why was it necessary to include `Point::?`
- What would happen if we removed `Point::?`

4/4/05

CS250 Introduction to Computer Science II

5

---

## Polymorphism in Action

- Write the statements needed to create objects of type Circle and Point
- Call the member function print on these objects
  - What is the output?
- Create a pointer to a Point object
- Create a pointer to a Circle object

4/4/05

CS250 Introduction to Computer Science II

6

## Polymorphism

- Assign the pointer to a Point object to the Point object created earlier
- Assign the pointer to a Circle object to the Circle object created earlier
- Invoke the print member function on both of these pointers
  - What is the output?

4/4/05

CS250 Introduction to Computer Science II

7

## Polymorphism in Action

- Point the pointer to a Point object to a Circle object
  - Is this possible?
  - Why?
- Invoke the print member function using this pointer
  - What is the output?

4/4/05

CS250 Introduction to Computer Science II

8

## Polymorphism in Action

- Assigning the address of a derived class object to a base class pointer is allowed
- The derived class is an object of its base class
- However the functions of the base class will be invoked rather than the members of the derived class

4/4/05

CS250 Introduction to Computer Science II

9

## Derived & Base Class Pointers

- Point the pointer to a Circle object to a Point object
  - Is this possible?
  - Why?

4/4/05

CS250 Introduction to Computer Science II

10

## Derived & Base Class Pointers

- The compiler does not allow you to assign a base class object to a derived class pointer
- Imagine if we could assign a Circle pointer to a Point object
  - What would happen if we invoked the setRadius member function?

4/4/05

CS250 Introduction to Computer Science II

11

## Summary

- We discussed the basics of polymorphism
- Next time we will talk about virtual functions
- We covered:
  - Pages 663 - 672

4/4/05

CS250 Introduction to Computer Science II

12