

Polymorphism II

- Polymorphism allows a derived class to substitute for a base class in variables and function calls.
- When a derived class object substitutes for a base class object (via copying), the copy of the object **loses all the abilities of the derived class.**
 - Extra fields are not copied from the derived class into the base class.
 - Extra methods can no longer be called on the new object (the copy).
 - Method calls on the new object (the copy) will always refer to the base class methods.

- Polymorphism also allows a **derived class pointer** to substitute for a **base class pointer**.
- Because when assigning pointers, no copies are made, derived class objects accessed through a base class pointer still retain all their extra abilities provided by the derived class.
- When an overridden method is called on a derived class object accessed through a base class pointer, two actions are possible.
 - If the method is **not virtual**, the **base class method** is called (same as if we were using regular objects rather than pointers to objects).
- If the method is **virtual**, the **derived class method** is called.

