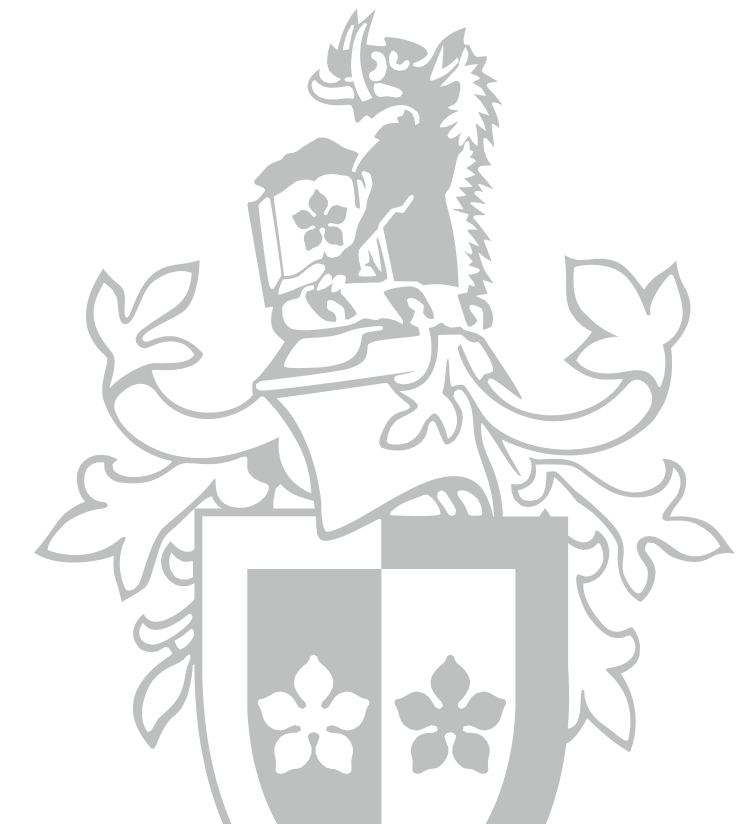


Polymorphism

Charlotte Pierce



SWIN
BUR
NE

SWINBURNE
UNIVERSITY OF
TECHNOLOGY

Use child objects where the
parent is expected

Refer to an object using any of the classes it **is a** kind of

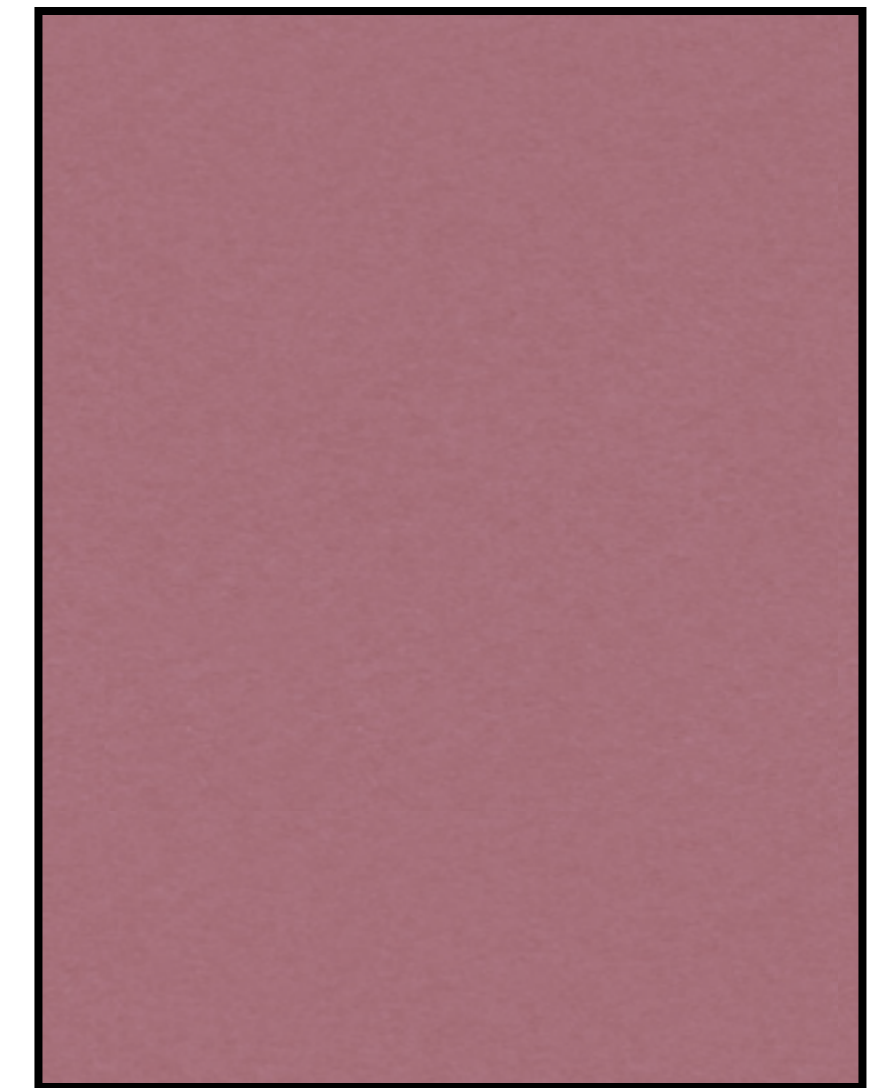
Object o Does o refer to an object?



Shape s Does s refer to a shape?



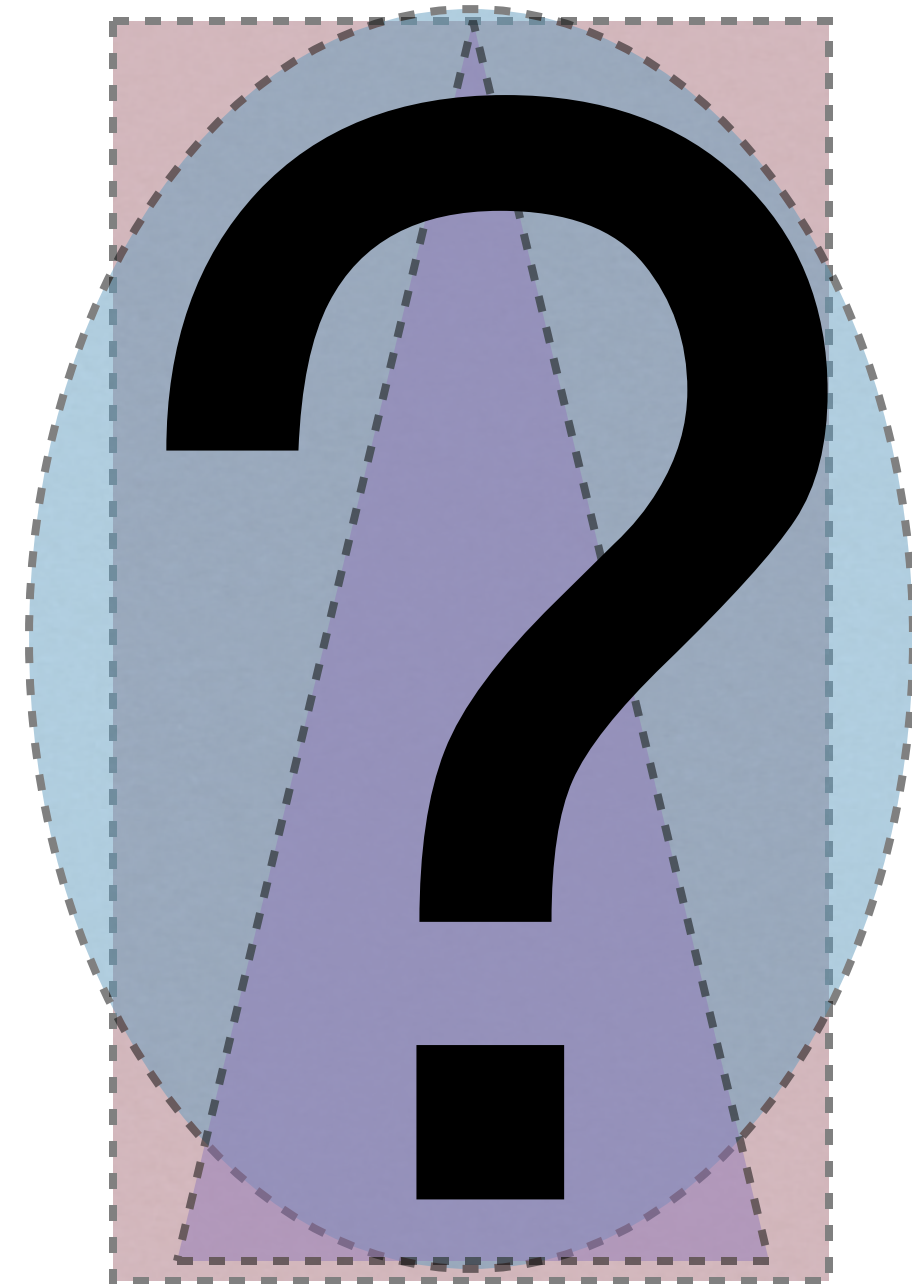
Rectangle r Does r refer to a rectangle?



Objects behave based on their **actual** class!

Shape s

What will be drawn?



This is called **polymorphism**

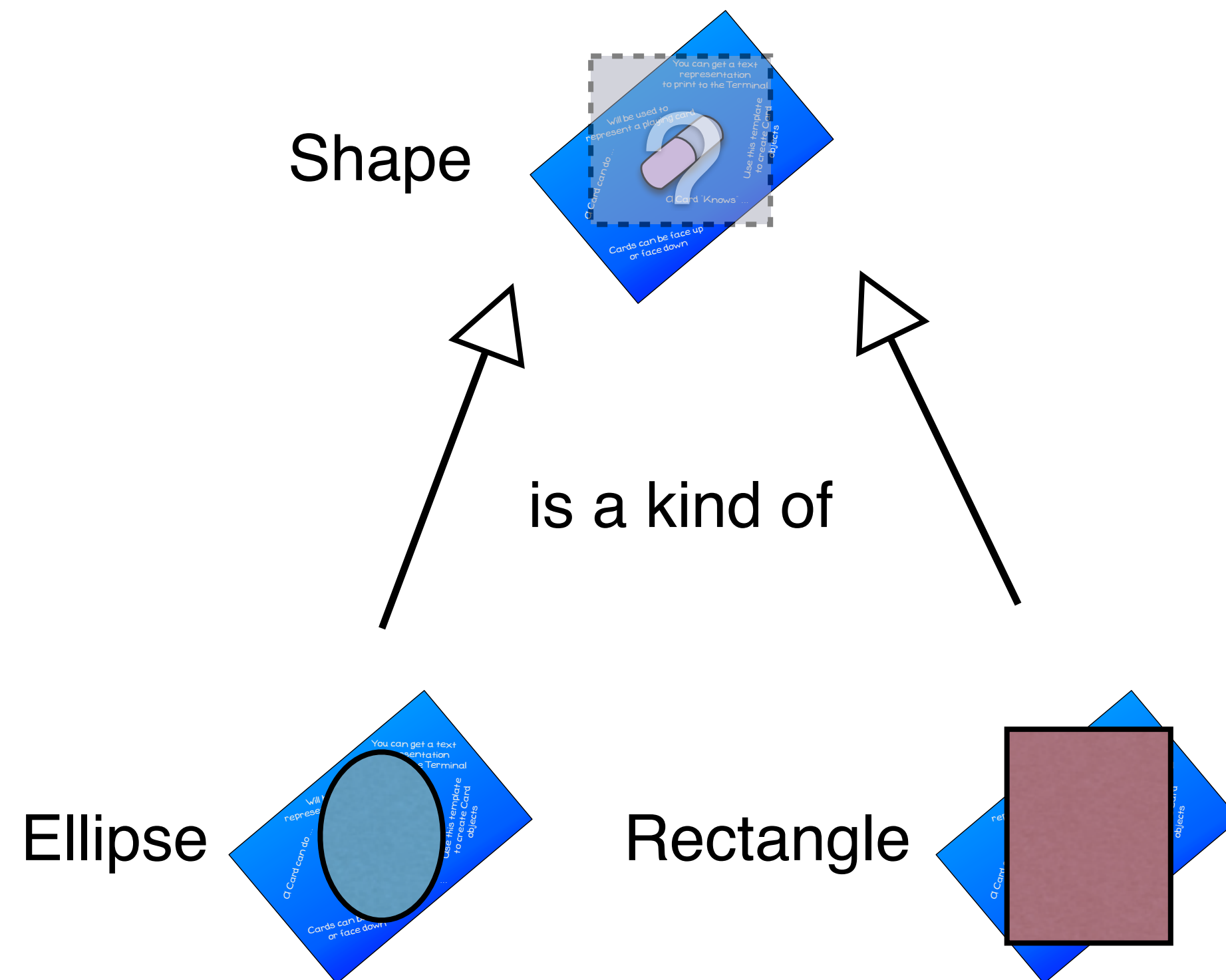
Poly

Morph

Many

Forms

Parent classes can have *placeholder* methods that **must** be overridden



How does Shape Draw?
It doesn't; Draw is a placeholder = **abstract**
abstract classes **cannot create objects**

Rectangle must override draw

Ellipse must override draw

Abstract methods of base classes

C++

virtual void draw () = 0;

C#

public abstract void Draw();

Java

public abstract void draw();

Objective-C

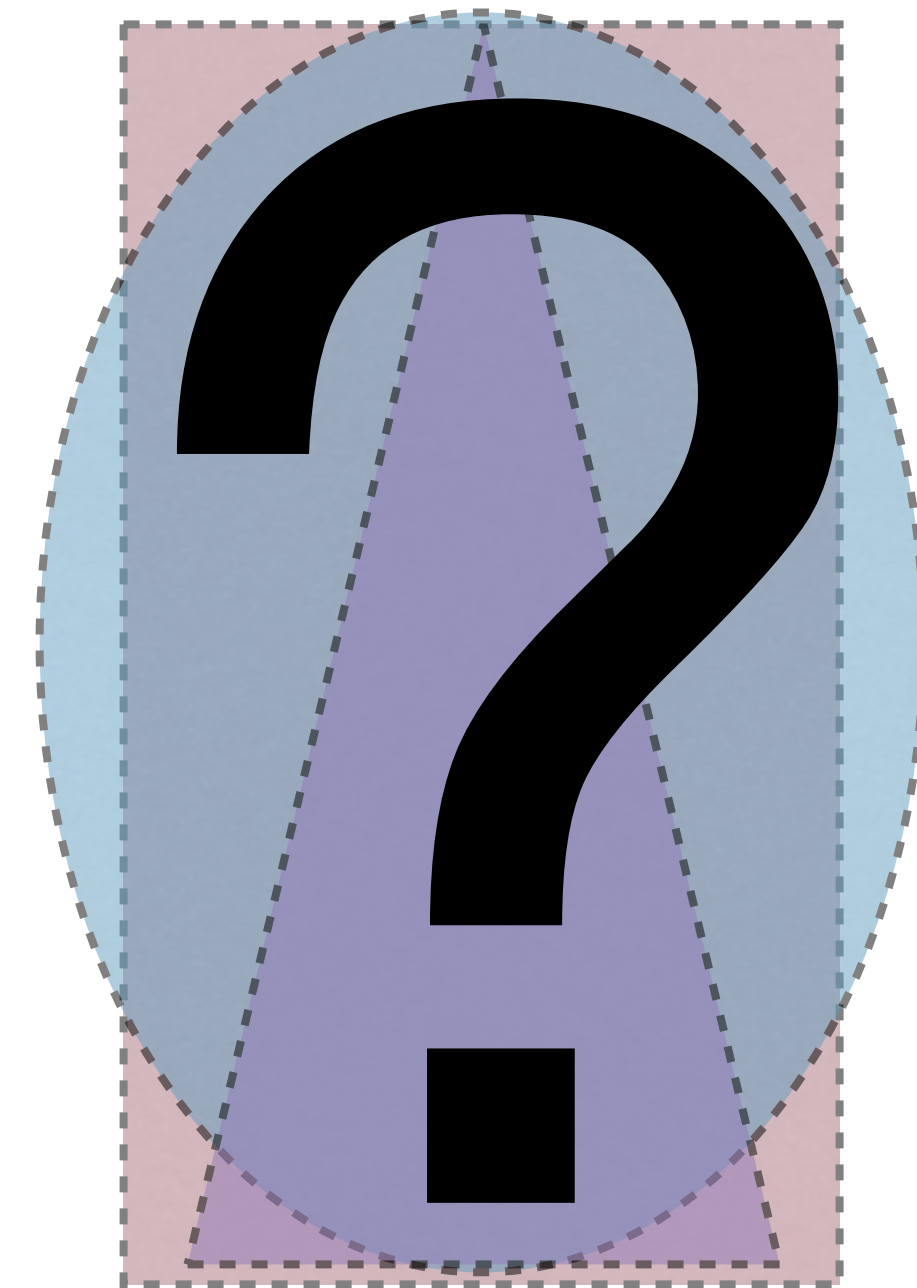
- (void) draw;

How do inheritance and
polymorphism help
development?

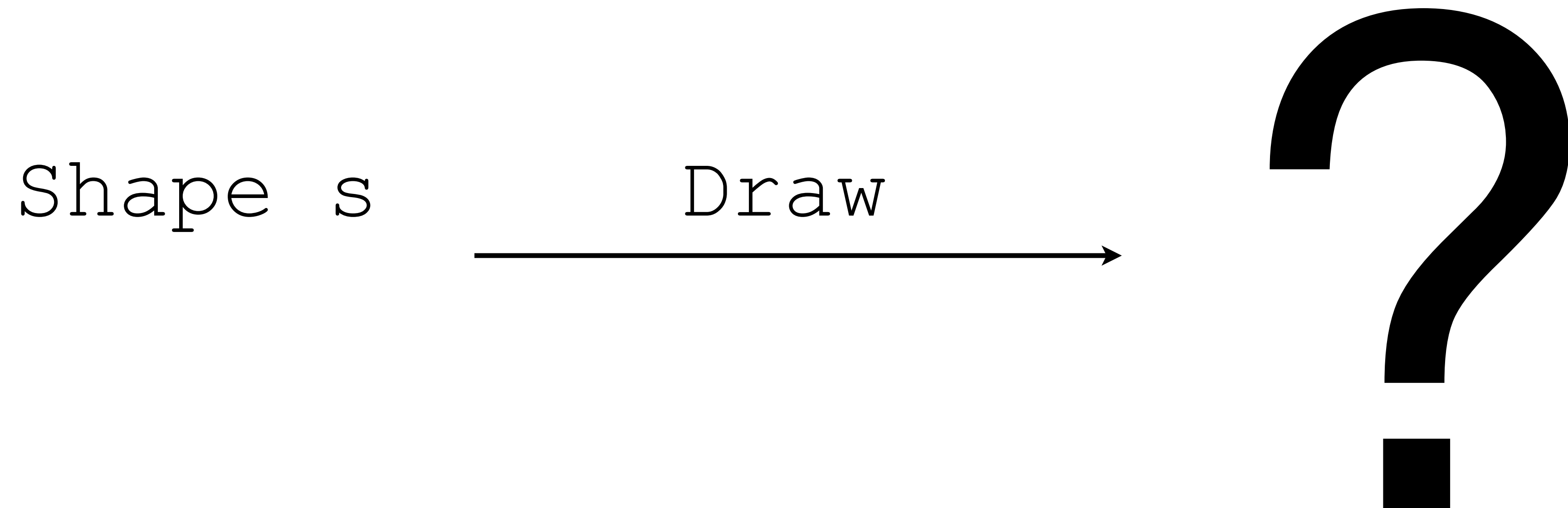
Flexibility: Refer to a parent class,
but get child objects... they work as
expected!

Shape s

Draw

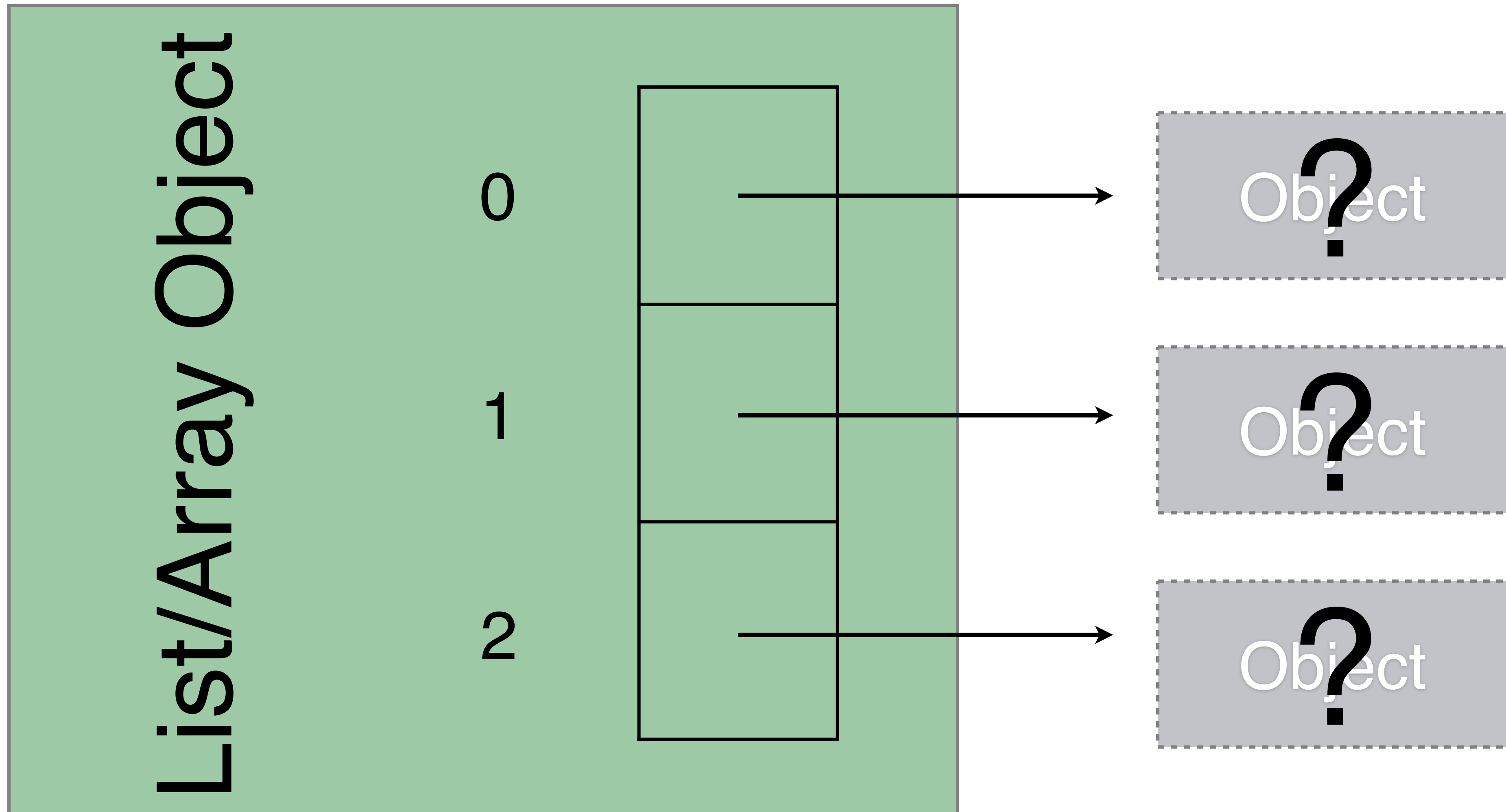


Extensible: add new children without needing to change uses



Can it only be Rectangle/Triangle/Ellipse?

Adaptable: Utilities like collection classes can work on Objects



Polymorphism helps bring
flexibility, extensibility, and
adaptability to your OO programs