

Summary 10 - Inheritance

- What is this?

If class B *extends* class A, we say B *is-a* A and B *inherits* all the fields, constructors and methods of A.

Thus, inheritance allows for *code re-use*, in that the *subclass* B can:

- use all of the functionality of the *superclass* A,
- add more functionality, and
- *override* functionality of the superclass as needed.

Consider the example to the right:

- SquareEye inherits from Eye how it responds to the mouse (among other things).
- SquareEye inherits from Eye the fields *eyeRadius* and *eyeBall* that are used in SquareEye's *paintComponent* method.
- SquareEye overrides the *paintComponent* method that is defined in Eye, drawing a square eye.

When you write *this.blah* in class X, Java looks for *blah* in class X first. If it doesn't find it there, it looks in the superclass of X. If not there, it looks in the superclass' superclass. And so forth.

The *Object* class is at the top of the hierarchy. All classes extend the Object class, either directly or indirectly.

When a method of superclass A is also defined in subclass B, we say that the method in B *overrides* the one in A. If the method in B wants to refer to the method in its superclass, it does so with the *super* keyword, as in the example to the right.

Similarly, a constructor in class B can refer to the superclass' constructor by using the *super* keyword like this:

```
super(...);
```

Any such call to the superclass' constructor must be the first statement of the constructor. If there is no such statement, Java inserts (invisibly) the statement

```
super();
```

- Example

```
public class SquareEye extends Eye {  
    @Override  
    public void paintComponent(Graphics graphics) {  
        super.paintComponent(graphics);  
  
        // Draw outline of eye.  
        graphics.setColor(this.eyeColor);  
        graphics.fillRect(0, 0, 2 * this.eyeRadius,  
                           2 * this.eyeRadius);  
  
        // Draw eyeball.  
        this.eyeBall.draw(graphics);  
    }  
}
```

- For further study:

- *Big Java*, Chapter 10 *Inheritance*
- This summary's *author*: David Mutchler