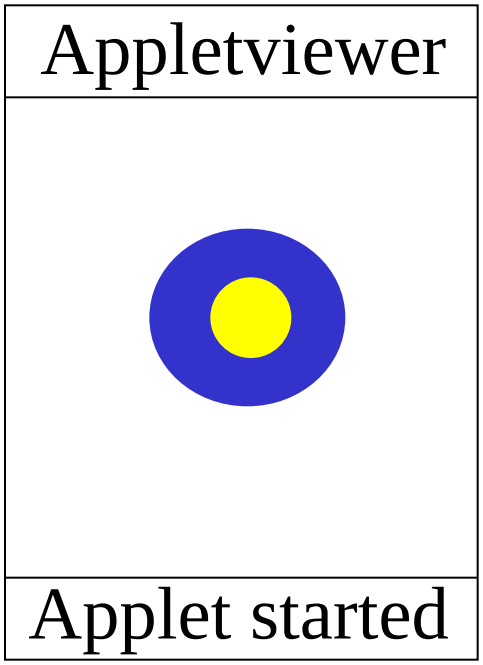
Step Into Java: Interfaces

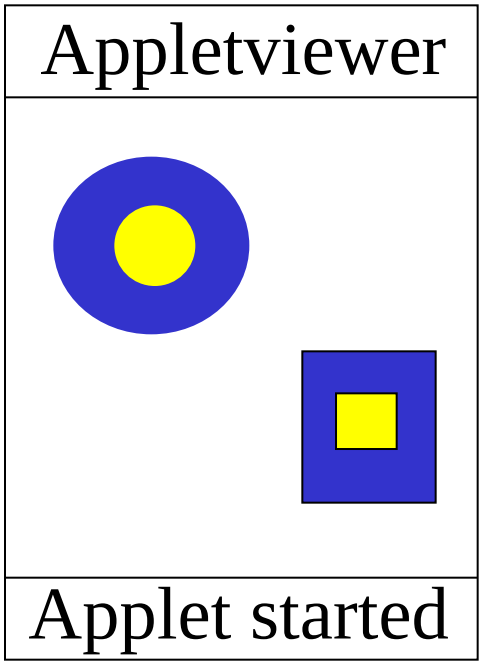
Mr. Neat

Java

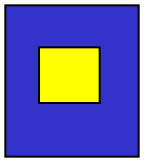
What if I wanted to drag this object around the window?



What if I wanted to drag either of these objects around the window?



We want to have
a variable in our program
that can hold objects from
different classes.



There are many ways to
do this in java.

```
public class WhatADrag extends WindowController
{
        private FilledRect box;
        private Location lastPoint;
        private boolean inBox;

        public void begin()
        {
                box = new FilledRect(30,30,50,50,canvas);
                box.setColor(Color.blue);
        }

        public void onMousePress(Location point)
        {
                lastPoint=point;
                inBox=box.contains(point);
        }
        public void onMouseDrag(Location point)
        {
                if(inBox)
                {
                        box.move(point.getX()-lastPoint.getX(),
                        point.getY() - lastPoint.getY());
                        lastPoint = point;
                }
```

We are going to explore *interfaces* to do this

What is an *interface?*

An interface is a specification of the methods that an object must support.

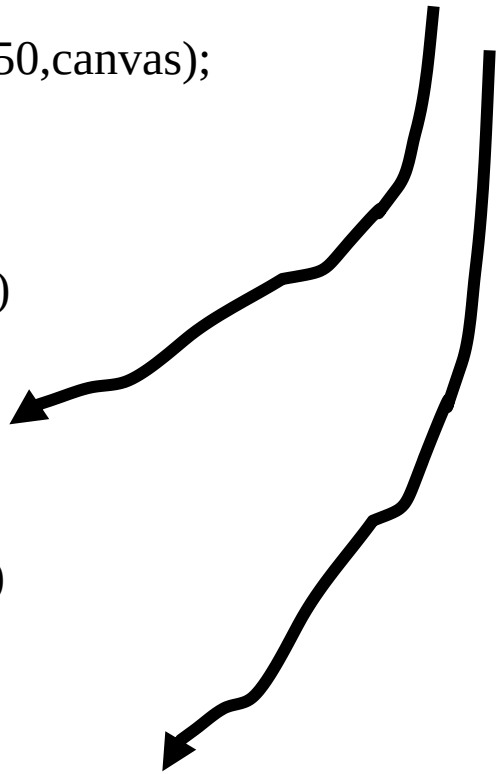
What does that mean???

```
public class WhatADrag extends WindowController
{
        private FilledRect box;
        private Location lastPoint;
        private boolean inBox;

        public void begin()
        {
                box = new FilledRect(30,30,50,50,canvas);
                box.setColor(Color.blue);
        }

        public void onMousePress(Location point)
        {
                lastPoint=point;
                inBox=box.contains(point);
        }
        public void onMouseDrag(Location point)
        {
                if(inBox)
                {
                        box.move(point.getX()-lastPoint.getX(),
                        point.getY() - lastPoint.getY());
                        lastPoint = point;
                }
```

What methods are used here?



Let's Look at One….

sort of like class

name of interface

```
public interface Movable
{
    public void move(double x, double y);
}
```

List of methods that must be defined in the class implementing the interface

don't forget

This allows us to make this declaration:

Movable someObject;

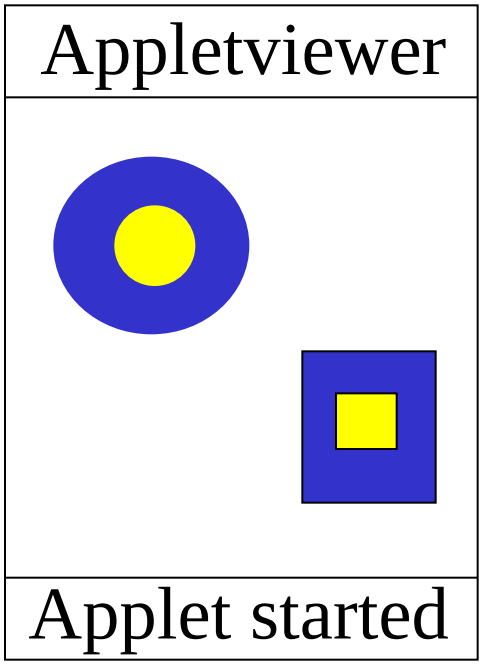
Any object referred to by someObject MUST have a move method as specified in the interface.

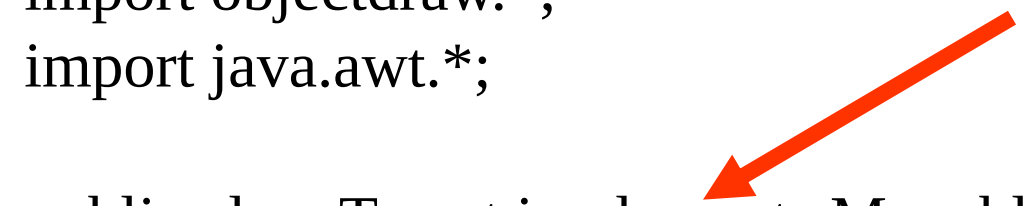
How do we associated objects of some class with an interface?

```
public class joe implements Movable
{

}
```

Back to the example





```
import objectdraw.*;
import java.awt.*;

public class Target implements Movable
{
        private int OUTERDIA = 30;
        private int INNERDIA = 15;

        public Target(double x, double y, DrawingCanvas neatcanvas)
        {
                outer = new FilledOval(x,y,OUTERDIA,OUTERDIA,neatc
                outer.setColor(OUTERCOLOR);
        }

        public void move(double dx, double dy)
        {
                outer.move(dx,dy);
                inner.move(dx,dy);
        }
```

Let's Look at One….

sort of like class

name of interface

```
public interface Movable
{
    public void move(double x, double y);
}
```

List of methods that must be defined in the class implementing the interface

don't forget

```
public class Square implements Movable
{
        private double OUTERSIDE=30.0;
        private int INNERSIDE=15;

        public Square(Location loc, DrawingCanvas neatcanvas)
        {
                Location in = new Location(loc);
                in.translate(7,8);
        }

        public void move(double dx, double dy)
        {
                outer.move(dx,dy);
                inner.move(dx,dy);
        }
```

```
public class WhatADrag2 extends WindowController
{
        private Target bullseye;
        private Square sbox;
        private Movable dragme;
        private boolean dragged;

        public void onMousePress(Location point)
        {
                lastPoint=point;
                if(bullseye.contains(point))
                {
                        dragme = bullseye;
                        dragged = true;
                }
                else if(sbox.contains(point))
                {
                        dragme = sbox;
                        dragged = true;
                }
                else
                {
                        dragged = false;
                }
        }
```

```
public void onMouseDrag(Location point)
        {
                if(dragged)
                {
                        dragme.move(point.getX()-lastPoint.getX(),
                        point.getY() - lastPoint.getY());
                        lastPoint = point;
                }
        }
```

Using *Interfaces*

- an interface declaration looks like a class declaration, except:
  - no constructors
  - no bodies of methods

Using *Interfaces*

- a class can implement an interface as long as the class provides public methods for all the methods listed in the interface

Using *Interfaces*

- a variable whose type is an interface may refer to objects from any class that implements that interface.

Next Lab Drag 2 Bodies

-drag a _____and your other new class objects around the window

- Use the new class defined in the previous lab

-Both _____objects and new class objects must implement a Movable interface