Review:
Global (Instance) Variables

Mr. Neat

Java

Review

Global Variables (Instance Variables)

Global variables are variables that are defined outside of any method, but inside of a class definition.

Global Variables (Instance Variables)

Every variable has scope. Its scope is dependent on where it is defined.

```
public static void main(String args[])
{
    // following line is necessary for onMouseClick, don't change

    MouseController mC = new MouseController(Canvas.getInstance(),new starter());

    // put your code here:

    Rectangle m = new Rectangle(50,100,300,300);
    m.draw();
}
```

scope for *m* is main() in this program

Global Variables (Instance Variables)

```
public class starter implements InputControl
{
	static Rectangle m;

	public static void main(String args[])
	{
		MouseController mC = new MouseController(Canvas.getInstance(),new starter());
		m = new Rectangle(50,100,300,300);
		m.draw();
	}
	public void onMouseClick(double x, double y)
	{
		m.translate(5.0,0.0);
	}
}
```

- move Rectangle *m* outside of the method, but in the class definition
- now the scope of *m* is the whole class
- both *main* and *onMouseClick* can use *m*
- note *static* is required since *main* is *static* (more later)

Global Variables (Instance Variables)

When defining a new variable, how do you decide what its scope is?

If only the method needs to know about it, define it within the method. But if another method needs to use it too or know about it, define it as a global variable.

Next Lab...

Make your labeled Rectangle object take a step each time the letter “d” is pressed