Step Into Java: this….continued

Mr. Neat

Java

this

- a key word
- a name for the object that is currently executing the method

*teen.java (user)

```
  joe = new Target(50.,50.,canvas);
  joe.moveTo(0,10);
```

Target.java (class creator)

```
public void move(double dx, double dy)
{
        outer.move(dx,dy);
        inner.move(dx,dy);
}
public void moveTo(double x, double y)
{
        this.move(x-outer.getX(), y-outer.getY());
        // move(x-outer.getX(), y-outer.getY());
```

What if we wanted to send moveTo a Location object rather than an x,y pair?

```
public void moveTo(Location pt)
{
    // blah, blah, blah
}
```

This is called overloading and we have seen it already with constructors.

Two methods with the same name, but different formal parameter lists.

```
public void moveTo(Location pt)
{
	this.move(pt.getX()-outer.getX(), pt.getY()-outer.getY());
}
```

We have seen this with constructors previously:

```
public Target(double x, double y, DrawingCanvas can)
{
        // code
}
public Target(Location loc, DrawingCanvas can);
{
        // code
}
```

If you did it, how did you do it?

```
public Target(double x, double y, DrawingCanvas nc)
{
        outer = new FilledOval(x,y,OUTERDIA,OUTERDIA,nc)
        outer.setColor(OUTERCOLOR);
        inner = new FilledOval(x+OUTERDIA/2 - INNERDIA/2
        inner.setColor(INNERCOLOR);
}
```

What would we do to overload this constructor for a Location object?

Another one liner….

```
public Target(Location loc, DrawingCanvas dc)
{
        this(loc.getX(),loc.getY(),dc);
}
```

“this” is how you call an existing constructor from a new constructor

Why is “this” a wise thing to do from a programming perspective?

Next Lab

- add a second constructor to your ___________ class
- use *this*
- verify your new constructor works