Objects

An object has
1. encapsulated data (knows, or remembers, stuff)
2. methods (can do stuff)

Object ← message(args)

In object-oriented programming, by sending
messages (requests) to objects

An object knows how to respond to specific messages,
(a) doing some stuff itself, and/or
(b) delegating some stuff to other objects.

E.g., A list `l` knows how to sort itself

```
l.sort() # l is the message; l is the object.
```

Another example: Graphics programming (2-D here).

```
import graphics

w = GraphWin() # Causes a graphics window to be displayed.
p = Point(50,50) # creates (but doesn't draw) a point at (50,50)
p.getX()
50
p.getY()
50
p.draw(w) # Ask p to draw itself on the graphics window, w.
w.close()
```

Graphics coordinates (x,y)
Origin (0,0) at top left

used by all graphics systems
- computers
- TV
- etc.
x and y are pixels.
```python
>>> from graphics import *
>>> w = GraphWin(300,300)
>>> p = Point(50,50)
>>> q = Point(100,100)
>>> ln = Line(p,q)
>>> ln.draw(w)

>>> w.close()
```

**Notation**

> p = Point(50,50)  # constructs an object of type Point,
# with location (50,50) and
# binds variable p to it.

```
Point
 x = 50
 y = 50
```

> p.getX()
50

```
<class names> ( <arguments> )

=> an object of type <class names>

Eq. point (50,50)

line (p, q)  # where p, q are Points

<object> . <method names> ( <arguments> )

Egs. p.getX()
p.draw(w)
```
# graphics1.py -- Simple graphics examples
from graphics import *

def main():
    win = GraphWin("My Shapes", 200, 200)
    p = Point(100, 100)
    p.draw(win)
    win.getMouse()  # Pause to view result

    c = Circle(p, 30)
    c.setFill('green')
    c.draw(win)
    win.getMouse()  # Pause to view result

    print "p.getX() =", p.getX()
    win.getMouse()  # Pause to view result

    t = Text(p, "Green")
    t.draw(win)
    win.getMouse()  # Pause to view result

    l = Line(Point(20, 150), Point(180, 150))
    l.setOutline('red')
    l.draw(win)
    win.getMouse()  # Pause to view result

    r = Rectangle(Point(20, 20), Point(40, 50))
    r.draw(win)
    win.getMouse()  # Pause to view result

    r.undraw()
    r.move(5, 0)
    r.draw(win)
    win.getMouse()  # Pause to view result

    r.undraw()
    r.move(5, 0)
    r.draw(win)
    win.getMouse()  # Pause to view result

    r.undraw()
    r.move(5, 0)
    r.draw(win)
    win.getMouse()  # Pause to view result

    r.undraw()
    r.move(5, 0)
    r.draw(win)
    win.getMouse()  # Pause to view result

    r.undraw()
    r.move(5, 0)
    r.draw(win)
    win.getMouse()  # Pause to view result

    r.undraw()
    r.move(5, 0)
    r.draw(win)
    win.getMouse()  # Pause to view result

    for i in range(100):
        r.undraw()
        r.move(1, 0)
        r.draw(win)

    win.close()  # Close window when done

main()