CSI31 Introduction to Computer Programming I

Dr. Sharon Persinger
Fall 2018

Topics

- Object-oriented programming
- Graphics module

Objects and object-oriented programming

- An object is an active piece of data that has both
  - Information: data, and
  - Operations or functions or methods.

Examples of objects

- Library system
  - Book object
    - Data: title, author, publisher, subject, checked out status, ...
    - Operations: change status, ...

Examples of objects

- Student record keeping system
  - Student object
    - Data: name, ID number, address, when enrolled, all courses, GPA, ...
    - Operations: create student, change address, add course, drop course, compute GPA, print address, ...

Examples of objects

- Graphics
  - Geometric objects: point, line, rectangle, triangle, circle
    - Data: location information, size, color, ...
    - Operations: create, draw, move, change color, ...

Object-oriented programming

- Two kinds of programming with objects
  - Later: Design and implement the objects.
  - Now: Use objects defined by another programmer.

Simple graphics with module graphics.py

Simple graphics with module graphics.py in Python shell

```
from graphics import *

w = GraphWin()
p = Point(50, 100) #Create Point object p
p.draw(w) #Draw the Point p
q = Point(100, 150)
q.draw(w)
```

Default coordinate system

More graphics

```
c = Circle(q, 20)
c.setFill('green')
c.draw(w)
label = Text(q, 'circle')
label.draw(w)
```

More graphics

```
p1 = Point(75, 75)
p2 = Point(80, 100)
rect = Rectangle(p1, p2)
rect.draw(w)
line = Line(Point(20, 20), Point(5, 200))
line.draw(w)
```

Graphics objects

Point object has data: x-coordinate, y-coordinate
Point object has methods, call them with dot operator syntax
constructor creates the object
p = Point(20, 50)
p.getX(), p.getY(), p.draw(w), p.move(dx,dy)

Object syntax

Create an instance of an object of a certain class with a constructor

‹class-name›(‹param1›, ‹param2›, ...)

Dot operator . Call the method that belongs to a particular object

Take care with copies!

```
leftEye = Circle(Point(80, 50), 5)
leftEye.setFill('yellow')
leftEye.setOutline('red')
leftEye.draw(w)
rightEye = leftEye
rightEye.move(20, 0)
```

Try it out and see what happens. Why?

Aliasing problem

Clone an object

You could create rightEye by writing code, but

```
rightEye = leftEye.clone()
rightEye.move(20, 0)
```

clone makes a separate independent copy

Write code to draw something simple -

stick figure

house

tree

Get graphics.py

- http://mcsp.wartburg.edu/zelle/python/