

Objects and Graphics

Rose-Hulman Institute of Technology

Computer Science and Software Engineering

Check out `05-ObjectsAndGraphics` from SVN. Get help if you're stuck.

Outline

- The object of objects
- Graphics
- Creating and using objects
- Interactive graphics
- In-class practice time

The object of objects

- Data types for numbers are passive
- Most modern computer programs are built using an *Object-Oriented* (OO) approach
 - An *object* is an active data type
 - It *knows* stuff
 - It *does* stuff

The object of objects

- Basic Idea of OO development
 - View a complex system as the interaction of simple objects
 - Example:
 - the human body is a complex system
 - the simulation of a character in the Sims is a complex system

Q2

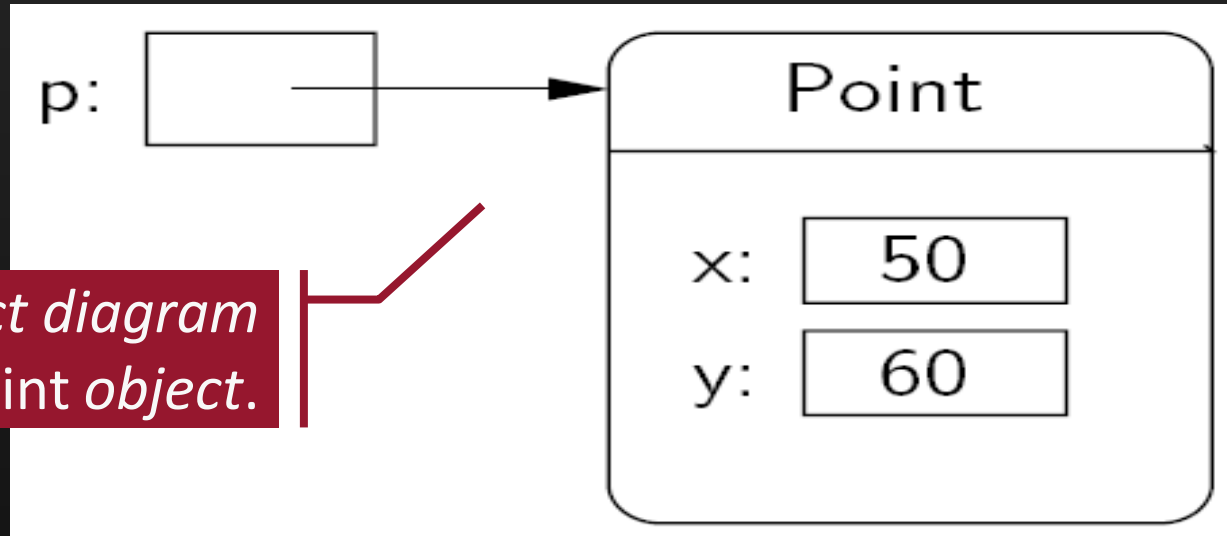
How do objects interact?

- Objects interact by sending each other *messages*
 - Message: request for object to perform one of its operations
 - Example: the brain can ask the feet to walk
 - In Python, messages happen via *method calls*.
- `win = GraphWin("Window", 10, 20)` **# constructor**
- `>>> p = Point(50, 60)` **# constructor**
- `>>> p.getX()` **# accessor method**
- `>>> p.getY()` **# accessor method**
- `>>> p.draw(win)` **# method**

Q3,4

How do objects interact? Point

```
p = Point(50, 60)
```



UML *object diagram*
for a *point object*.

Q5

Simple graphics programming

- Great way to learn about objects
- *Computer Graphics*: study of graphics programming
 - Important for gaming and movie industries
 - Military applications
 - Is fun
- Graphical User Interface (GUI)

Review: Two Ways to import

- Must import graphics library before accessing it
 - `>>> import zellegraphics`
 - `>>> win = zellegraphics.GraphWin()`
- Another way to import graphics library
 - `>>> from zellegraphics import *`
 - `win = GraphWin()`

Graphics window

- Collection of tiny points called pixel
 - Pixel: picture element
 - Has a title, length, and width
 - E.g. height = 200 pixels, width = 200 pixels
 - How many pixels?
- Computer monitor
 - # pixels wide
 - # pixels tall

Using graphical objects

- Look at the **alienFace** module in today's project



Q6

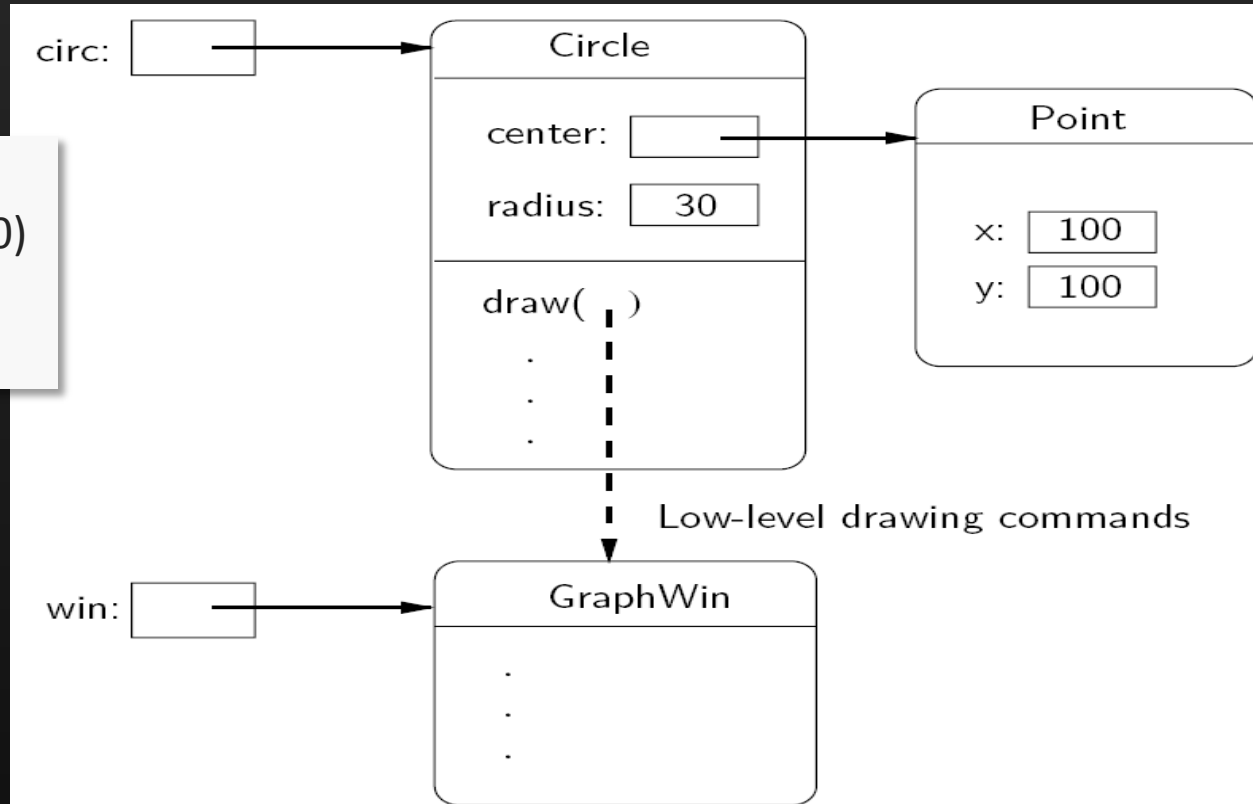
Recap: Class and object terminology

- Different **types** of objects
 - Point, Line, Rectangle, Oval, Text
 - These are examples of *classes*
- Different *objects*
 - head, leftEye, rightEye, mouth, message
 - Each is an **instance** of a class
 - Created using a constructor
 - Objects have instance variables
 - Objects use methods to operate on instance variables

Q7, 8

Object interaction to draw a circle

```
from zellegraphics import *  
circ = Circle(Point(100, 100), 30)  
win = GraphWin()  
circ.draw(win)
```



Interactive graphics

- *GUI—Graphical User Interface*
 - Accepts input
 - Keyboard, mouse clicks, menu, text box
 - Displays output
 - In graphical format
 - On-the-fly
- Developed using *Event-Driven Programming*
 - Program draws interface elements (widgets) and waits
 - Program responds when user does something

Example: getMouse

- *win.getMouse()*
 - Causes the program to pause, waiting for the user to click with the mouse somewhere in the window
 - To find out where it was clicked, assign it to a variable:
 - *p = win.getMouse()*

Q10-12

Mouse Event Exercise

- Create a program in module, **clickMe**, with a window labeled “Click Me!” that displays the message **You clicked (x, y)** to the console the first 5 times the user clicks in the window.
- The program also draws a red-filled circle, with blue outline, in the location of each of these first 5 clicks.
- The program closes the window on the 6th click