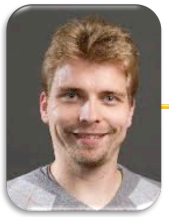


UC Berkeley EECS
Adj. Ass. Prof.
Dr. Gerald Friedland

Computational Structures in Data Science

Lecture #11: Object-Oriented Programming II

Forget your old alarm system. This drone will protect your house.

http://money.cnn.com/2016/11/03/technology/drone-home-alarm-system/index.html

March 23rd, 2018

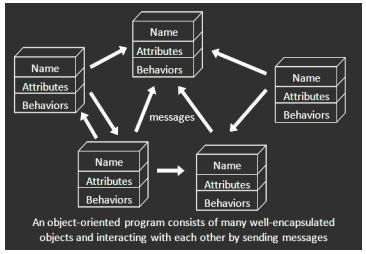
http://inst.eecs.berkeley.edu/~cs88

Computational Concepts Toolbox

- Data type: values, literals, operations,
- Expressions, Call expression
- Variables
- Assignment Statement
- Sequences: tuple, list
- Dictionaries
- Data structures
- Tuple assignment
- Function Definition Statement
- Conditional Statement
- Iteration: list comp, for, while
- Lambda function expr.
- Higher Order Functions
  - Functions as Values
  - Functions with functions as argument
  - Assignment of function values
- Higher order function patterns
  - Map, Filter, Reduce
- Function factories – create and return functions
- Recursion
  - Linear, Tail, Tree
- Abstract Data Types
- Generators
- Mutation
- Object Orientation

Object-Oriented Programming (OOP)

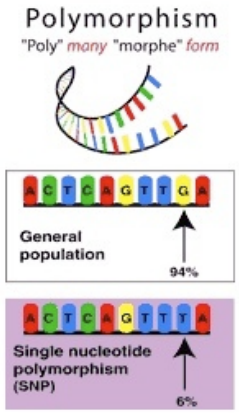
- Objects as data structures
  - With methods you ask of them
    - » These are the behaviors
  - With local state, to remember
    - » These are the attributes
- Classes & Instances
  - Instance an example of class
  - E.g., Fluffy is instance of Dog
- Inheritance saves code
  - Hierarchical classes
  - E.g., pianist special case of musician, a special case of performer
- Examples (tho not pure)
  - Java, C++



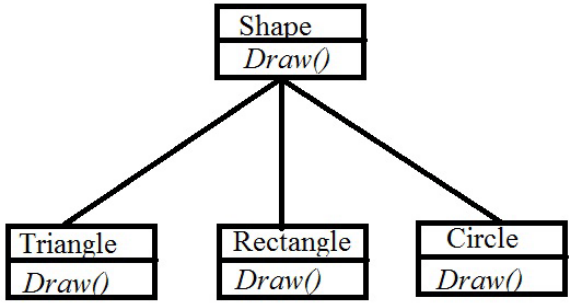
www3.ntu.edu.sg/home/ehchua/programming/java/images/OOP-Objects.gif

Polymorphism

- Different classes in the inheritance tree: Same method name, but different implementation



Polymorphism



- **In Python, Polymorphism is implemented by method overloading**

Polymorphism: Most known Example!

- **print!**

```
print (1)
print ("")
print ([])
print ({})
print (())
print (object)
print (sys)
```

- **Remember: Everything in Python is an Object!**

Example!

```
class Root:
    def draw(self):
        # the delegation chain stops here

class Shape(Root):
    def __init__(self, shapename, **kwds):
        self.shapename = shapename
        super().__init__(**kwds)

    def draw(self):
        print('Drawing.  Setting shape to:', self.shapename)
        super().draw()

class ColoredShape(Shape):
    def __init__(self, color, **kwds):
        self.color = color
        super().__init__(**kwds)

    def draw(self):
        print('Drawing.  Setting color to:', self.color)
        super().draw()

cs = ColoredShape(color='blue', shapename='square')
cs.draw()
```


Polymorphism

- **Let's try it…**
  - **Define a class**
  - **Inherit**
  - **Show polymorphism**
  - **Use super**

  - **Answer questions from class**