

Python Classes

I. Introduction: What you already (should) know

1. How to assign to variables (incl. Strings, ints, lists, dicts)

```
data = 'AGGTCTC'
```

2. How to create functions using the *def* command

```
def check_sequence(seq):
```

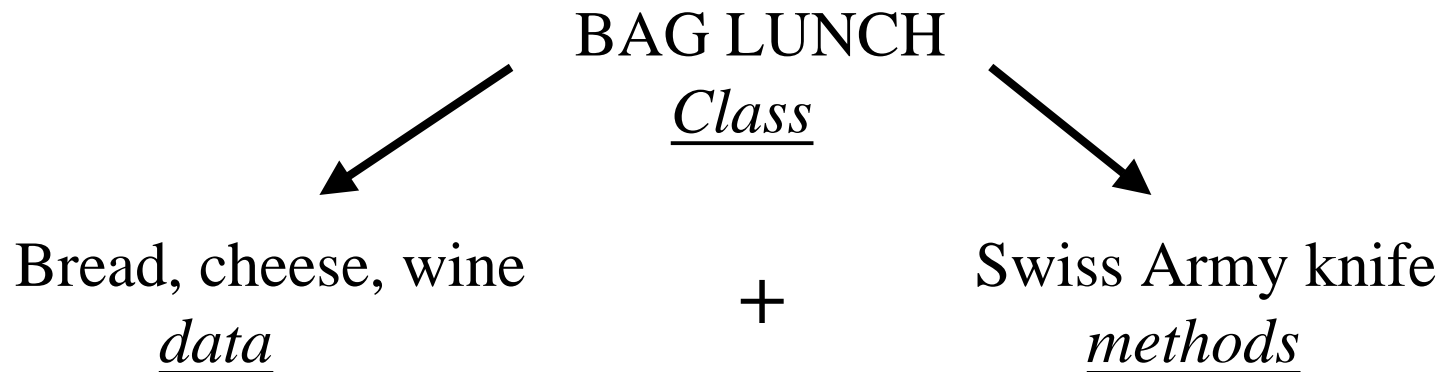
3. How to import modules and use their methods

```
import string  
string.lower(data)
```

II. Why use classes?

1. Classes provides means to organize *logic & data*
2. Classes allow code to be easily reused

Analogy



III. Class Syntax & Using Classes


```
class <name>:                                #assign to name
    data = value                             #shared class data
    def method(self, ...):                  #methods
        self.data = value                  #per-instance data
```

```
class SeqClass:
    data = ''
    def store_sequence(self, sequence):
        self.data = sequence
```

```
-----filename: Prot.py-----
class Proteins:
    universal = {'UUU': 'V', ...} #universal genetic code
    def return_aa(self, codon):
        """Returns amino acid given codon string"""
        aa = self.universal[codon]
        return aa
```

```
-----
>>> import Prot #just like string, re
>>> proteins = Proteins() #create instance
>>> proteins.universal #instance attribute
{'UCU': 'A', 'UUU': 'V'}
>>> proteins.return_aa('UUU') #call instance method
'V'
>>> prot_seq = proteins.return_aa('UUU')
>>> prot_seq = prot_seq + proteins.return_aa('UCU')
>>> prot_seq
'VA'
```

IV. More Examples

```
class FirstClass:   
    def __init__(self):  
        """Instance data"""  
        self.data = []  
    def storedata(self, value):  
        self.data.append(value)  
    def display(self):  
        print self.data
```

```
>>> execfile('Firstclass.py')  
>>> x = FirstClass()  
>>> x.storedata(5)  
>>> y = FirstClass()  
>>> y.storedata('spam')  
>>> x.display()  
[5]  
>>> y.display()  
[ 'spam' ]
```

```
>>> x.data.append(100)  
>>> x.display()  
[5,100]
```

