Object-Oriented Programming

Historical Perspective
Object-oriented programming was a huge conceptual change in 80s and 90s.

Simula, Algol, Smalltalk were early O-O languages

C++ was born from the language‘C’

Java was born

Grouping Data
Our brains can only handle 7 +/- 2 things at a time.

We must create abstractions and think hierarchically.

Function allow us to abstract sequence of operations.

We also need to abstract groupings of data, something we call ‘complex data types’ or ‘classes’.

Consider a social network site. What data might be recorded for each person in the system?

Other examples of grouping data:

Google result
A color

How about the Mastermind program? What data did we use that could have been grouped but wasn’t?

Grouping Data in Python

How would you represent a person in Python?

‘Lists’ don’t really help here as you have related but heterogeneous data. You want to name each part, e.g., person.lastName, person.firstName, etc.
**Python Classes**

The Class construct allows a programmer to define custom types (*programmer-defined types*).

Similar syntax to function def, but it’s a class definition. Define your fields by setting them to initial values.

```python
class Person:  # class definition
    name=""
    age=0
    friends = []
```

Note that the variables are being set outside any function!

Once you define a class, you can create *instances* of it. We call these instances *objects*. Create them with an object creation statement, set them using dot notation.

```python
# main
joe = Person()      # this creates a Person object with name="" and age=0
bill = Person()     # this creates a Person object with name="" and age=0
print joe.age       # prints 0
joe.age=34
joe.name="Joe Williams"
bill.name="William Joes"
bill.age=22
```

Formally, object creation statements are of the form:

```python
<object> = <nameOfClass>()
```

Field assignment statements use "." to specify which field:

```python
<objectName>,<fieldName>
```

**Terminology**

Person is a class.

joe and bill are instances of the class Person.

joe and bill are objects of type Person.

name and age are fields (also known as *data members*) of the class Person.

A class can also have functions defined within it (we’ll revisit this later).
In-Class Problems 1

1. Write a class 'Coordinate' with two fields, x and y, which are initialized to zero.

2. Write a function 'distance' which accepts two coordinates as parameters and returns the distance between them. This function should NOT be object-oriented (define it after the class definition).

3. Write a main program to test your class and function-- it should create two coordinates, call distance, then print the coordinates and the distance.

To do the math, use Python's Math Module and in particular: ‘sqrt’, and ‘pow’.

Class functions (methods)
In the discussion above, we discussed programmer-defined types and used Python classes to represent them.

We didn't really write an object-oriented function.

In the world of object-oriented programming, a class is really two things:

    class =  data fields  + functions that work on that data

In other words, we want to group data along with the specific functions that access and manipulate that data.

To do this, we define functions within the class construct:

    class Coordinate:
        x=0
        y=0
        def distance(self,point2):
            # code for distance

Then we call the functions using the form:  <object>.<function>

For example,

    point1 = Coordinate()
    point1.x=4
    point1.y=7
    distance = point1.distance(point2)

This is an object-oriented function call. Method is another name for function.
The first parameter is usually, by convention, called self. Self refers to the object on the left-hand side of the "." in the object-oriented function call.

There are really still two parameters sent to the function distance.

**In Class Problems 2**

1. Copy your coordinate program into another file.

2. Rewrite the code so that distance is a method inside the Coordinate class. You'll need to rewrite the main so that distance is called in an object-oriented manner.

3. Add a method 'equal' which returns true if two coordinate objects have the same x and y values. Write a test for this method and show how 'equal' is different from '==='.