Objects and Classes

Review: Objects

- In Java and other Object-Oriented Programming (OOP) languages, the focus is on ***objects***
- ***Objects*** are program modules that can do actions or be acted upon by other objects
- All objects have
  - ***Properties***
    - These are the *data* about an object
    - In Java we call them ***attributes*** or ***fields*** or ***instance variables***
  - ***Behaviours*** (***actions***)
    - In Java they are implemented as ***methods*** (more specifically, ***instance methods***)

Review: Objects and Classes

- Every object belongs to a specific ***class***
  - Objects that belong to the same class have the same properties and can perform the same actions
- We can think of a class as being a ***template*** or ***pattern*** or ***model*** or ***definition*** for objects of that class

Review: Object-Oriented Programming

- ***Object-oriented programs*** consist of ***interacting objects***
  - Objects are ***defined by*** classes
  - Objects can be ***created by*** objects of other classes (***client classes***) which ***use*** them in implementing a programming solution to a problem

Example: Social Networking

- Suppose we want to keep track of social contact information for our friends / relatives
- We wish to write a program that allows us to add contact information of a friend to our list of friends, remove a contact from the list, and print information about all our contacts.

Example: Social Networking

- Part of OOP design is deciding on what classes we will need for our problem
- Let's start with a class called Person, that will model the information about one person in our social network

Review: Class Definition

- A ***class definition*** consists of
  - Attribute declarations (also known as fields or instance variables)
  - Constructor definitions
  - Method definitions
- A class definition is stored in a file
  - With the same name as the class
  - With a .java extension on the file

Example: Person Class

- ***Attributes (instance variables, fields)***
  - What kind of information do we want to have about a person? Let’s keep it short for now
    - Person's name
    - Email address
  - What type should each of these be?
    - A name can be a string
    - An email address can be a string

Example Python: Person Class

```
class Person:

    def __init__(self, firstName=“”, lastName,=“” email=“”):
      self.firstName = firstName
      self.lastName = lastName
      self.email = email
```

- Note in Python we can assign default values to the attributes in this case we used an empty string

Example Java: Person Class

```
public class Person{
/* Attribute declarations */
    private String lastName;
    private String firstName;
    private String email;
```

- Why are the attributes private?
- Note that the instance variables are just being ***declared*** here (not explicitly assigned values)