What does Object-Oriented mean?

- Very common method of organizing code
  - Design classes, which encapsulate state and behavior
  - Some classes can be similar to, but different from their parent class
    - Super class, subclass
  - Inherit behavior, use as is or modify and use or both

- Complex to design a hierarchy of classes, but important
  - More of this in CompSci 108 or on-the-job training
  - We’re solving simple problems, not designing re-usable libraries
- Simple does not mean straight-forward, but not MS Vista!

Inheritance and Interfaces

- Interfaces provide method names and parameters
  - The method signature we can expect and thus use!
  - What can we do to an ArrayList? To a LinkedList?
  - What can we do to a Map or Set or PriorityQueue?
  - `java.util.Collection` is an interface

- Abstract classes can implement core, duplicated code
  - If we can add one object to a [set,map,list], can we add an entire list of objects? `java.util.AbstractCollection`
  - If we can iterate can we remove? Convert to array? Obtain size?

Model View Controller, MVC

- Very common Design Pattern
  - Capture common solutions to problems in a context
  - Forces that indicate when to employ the solution
    - Iterator, Composite, Decorator seen in CompSci 100

- Model has all the state and knows when it changes
  - Communicates changes to views (via controller)
  - Must be initialized, updated, etc.
  - Views can be GUI or otherwise, more than one!
  - Change spread-sheet, charts and formula change

Eugene (Gene) Myers

- Lead computer scientist/software engineer at Celera Genomics, then at Berkeley, now at Janelia Farms Research Institute (HHMI)

- “What really astounds me is the architecture of life. The system is extremely complex. It’s like it was designed.” ... “There's a huge intelligence there.”

- BLAST and WG-Shotgun
MVC Example, key-word-in-context

- User loads file
  - Where? Communicate to?
  - What changes in model?
  - What happens in view?

- User chooses word
  - First choice causes action once
  - Alternatives?
  - Generate context, display
  - How to show in any view?

Key Word in Context Explained

- For every different word, store where it occurs
  - *love* is the 1st, 3rd, 50th, and 1237th word in the file

- This data is kept in a map, key is word, value is ??
  - How do we generate the data in the map?

- Given a word, how do we find its context? How do we format?
  - All words are in an array, in order
  - Memory concerns?
  - Original KWIC paper by Parnas as comparison

KWIC main program/class

```java
public class ContextMain {
    public static void main(String[] args) {
        IModel model = new ContextModel();
        SimpleViewer view = new SimpleViewer("Compsci 100 KWIC", "word");
        view.setModel(model);
    }
}
```

- What changes in above for Markov assignment?
  - How can view communicate with any model?
  - View doesn’t change, model does!

Convention Summary

- Classes start with capital letter and then we have:
  - They’re public, except nested class? Protected means ...
  - camelCaseForMethods and ForClasses
  - Ivars, fields, instance variables, mySize, myMap, ...
  - Constants (public static) are ALL_CAPS

- Interfaces are IModel, IView, and so on
  - Not true for standard Java classes, yes for Compsci 100
  - Don’t need to label methods as abstract, but can

- Supply AbstractDefault implements IThing
  - Constructor, some state, some common behavior: extend!