Review of Java

- Classes are object factories
  - Encapsulate state/data and behavior/methods
  - Ask not what you can do to an object, but what ...

- A program is created by using classes in libraries provided and combining these with classes you design/implement
  - Design classes, write methods, classes communicate
  - Communication is via method call
Control in Java Programs

- We've concentrated on control within and between methods
  - Data types: primitive, array, String
  - Control: if, for-loop, return

- What are alternatives to "execute the next statement"?
  - Repeat this block
  - Execute this block when this condition holds
  - Return from this method
Smallest of 2, 3, ..., n

- We want to print the lesser of two elements, e.g., comparing the lengths of two DNA strands

```java
int small = Math.min(s1.length(), s2.length());
```

- Where does min function live? How do we access it?
  - Could we write this ourselves? Why use library method?

```java
public class Math {
    public static int min(int x, int y) {
        if (x < y) return x;
        else return y;
    }
}
```
Generalize from two to three

- Find the smallest of three strands: s1, s2, s3

```java
int small = ...
```

- **Choices in writing code?**
  - Write sequence of if statements
  - Call library method
  - Advantages? Disadvantages?
Generalize from three to N

- Find the smallest strand length of N (any number) in array

```java
public int smallest(String[] dnaCollection) {
    // return shortest length in dnaCollection
}
```

- How do we write this code? Where do we start?
Static methods analyzed

- Typically a method invokes behavior on an object
  - Returns property of object, e.g., `s . length()`;
  - Create new object: `s . substring(2, 5)`;
  - Object changes state: `dna . cleave(rna)`;

- Sometimes we don't need an object, e.g., square-root, min, even find CG ratio!
  - Invoked using class-name, not object
  - All information passed in, no internal state
  - Compare to String substring, need state of internal chars
How do we know about stuff?

- **Where is documentation for Math class?**
  - Where does Math class live in relation to other classes?
  - How do we access and read documentation?

- **By convention Java classes include comments processed by a program called javadoc that generates web pages**
  - Writing stylized comments facilitates browsable docs
  - API is application programming interface
Organization of classes

- **Java classes are organized into packages**
  - Keep related classes together
  - Facilitates conceptual use and development (from client/programmer view and developer/programmer view)

- **Access to classes provided by import statement**
  - All classes in java.lang imported silently
  - Math, String, Object, System,…
  - Other packages require providing compiler with location
    - Packages organized hierarchically and conventionally named

```java
import java.util.Arrays;   // to sort arrays
import org.biojava.bio.seq.DNATools;
```
Richard Stallman

- One of world's best programmers/hackers
  - Difference? Pejorative?
- Developed GNU software
  - C/C++, emacs, libraries
  - Basis for Linux
- Awards:
  - Macarthur genius award
  - Grace Murray Hopper
- Invented copyleft, free software
  - Free speech, not free beer
  - Basis for most bioinformatics tools, Perl, biojava, ...
What is an ORF?

  - From BD: look at AE008569 (see page 160)

- **Why are the six open reading frames?**
  - How do we know which is the best ORF?
  - Why does graphical representation help?
  - What about Java programming to find them
    - Why do we do this when it's already done