

# Using Objects (cont.) / Reading input

- Finish up String discussion from last time
- Review of object references
- Java API and packages
- using the Java API documentation
  - packages
  - method headers
- Ex of creating and using objects: reading input
  - Scanner class

# Announcements

- Claire's office hours: Tu 12 – 2; Th 10:30-12:30
- TA/Course producer office hours coming soon to web page.
- Do prog. experience survey today
- 1<sup>st</sup> prog. assgt. (PA1) to be published in the next few days
- Students in the following categories need to see me after class (or in office hours today):
  - missed the first lecture
  - no programming experience
  - not officially enrolled (e.g., on waiting list)

# Review of object references / String

# Java API

- API = Application Programmer Interface
- all the classes in the extensive Java library
  - Includes 1000s of classes
- We'll discuss further:
  - library organized into *packages*
  - how to find out how to use the classes provided

# Java API packages

- *packages* organize sets of related classes
- Examples:
  - `java.awt`  
(has some graphics classes such as **Rectangle**),
  - `java.lang`  
(has basics such as **String** and **System**)
- to use a class, need to know the package it is in
- import statement:  
`import java.awt.Rectangle;`

# Java API documentation

- API doc.
  - available on the web (linked from course web page)
  - can get a local copy (install JDK)
  - selected classes/methods in textbook (Appendix)
- API doc describes the *interface* for the classes:
  - class name + description
  - package name
  - *method headers* + descriptions

# What is a method header?

- Like function headers, but appear inside a class definition (we'll see class def next lecture)
- Example method call:  
**rect.translate(5, 10);**
- Corresponding method header:

**public void translate(int x, int y)**

# Java API Documentation

- Let's check out the API documentation on the web...



# Reading console input

- use `java.util.Scanner`
- Pass `System.in` to scanner constructor:  
`Scanner in = new Scanner(System.in);`
- Skips over whitespace, and reads in next sequence of non whitespace chars:  
`in.nextInt()`  
`in.nextDouble()`  
`in.next() [String]`
- Reads in whole line as a string:  
`in.nextLine()`

# Example: Reading input

- Small example program in **ReadName.java**

# **nextLine**

How do you write code that combines word-by-word reading (e.g., with **nextInt()** or **next()**) and line-by-line reading (i.e., with **nextLine()**)?

- Specific example: write code that when run does...

**Enter your age: 32**

**Enter your whole name: *Joseph P. Blow***

See **TestReadLine.java**