CS1110 lecture 4  9 Sept. Customizing a class & testing

- Classes: fields; getter & setter methods (Secs. 1.4.2 & 3.1 (pp. 105–110 only))
- Constructors, Sec. 3.1.3 (pp. 111–112)
- Testing methods, Appendix I.2.4 (p. 486)

Organizational tip 8652:
For classes with a lot of handouts (like CS1110), get a 3-ring binder and a 3-hole punch. Punch holes in the handouts and store them in the binder; this makes accessing them much easier. You can easily interleave other notes and papers, too.

A “must see” about academic integrity (on youtube): http://tinyurl.com/35HtF4n

One-on-One Sessions (optional)
Next 1.5 weeks, we are holding 1/2-hour one-on-one sessions on a computer with each student in CS1110.

Purpose: Help you develop a class as preparation for A1, give you a chance to ask questions. Not counted in final course grade.

Sign up on the course CMS (http://cms.csuglab.cornell.edu): Click on assignment One-on-one, find the schedule of times/instructors. Choose one.

Bring to the 1-on-1: the book, laptop w. DrJava if you have one.

Students with little or no programming experience report that these sessions are extremely helpful!

Office hours: [URL]
Already started: Prof Gries & Prof Lee (Tu/Th 10:10-10:55 Hollister 202), consultants TAs: will begin the week of Sept. 20, check the URL then

Field: a variable that is in each folder of a class.

```java
public class Worker {
    /** An instance is a worker in a certain organization. */
    private String lname; // Last name ("" if none; never null)
    private int ssn; // Social security #: in 0..999999999
    private Worker boss; // Immediate boss (null if none)

    Declarations of fields
    
    **/ An instance is a worker in a certain organization. */
    public class Worker {
        private String lname; // Last name ("" if none; never null)
        private int ssn; // Social security #: in 0..999999999
        private Worker boss; // Immediate boss (null if none)
    }

    Usually, fields are private, so methods that are outside the class can’t reference them.
    (Exception: private fields are accessible in the DrJava Interactions pane)
```

Quiz 2 on Tuesday 14 Sept
Purpose of a constructor (slide 6); Evaluating a new expression (slide 8)

Assignment A1 out today, due Sat., 18 Sept. on the CMS. Submit A1 earlier if you can so that we can start the iterative feedback process going.
Labs and one-on-ones (schedule yours on CMS) will help you with it.

Collaboration rules for assignment A1
• Work alone or with one partner — partners “group themselves” on the CMS well before submission; only one person submits the files.

Partners must do the work together, sit next to each other, with each taking turns “driving” (handling the mouse and keyboard). It is against the rules for one partner to develop code and later show it to the other.

• Never look at someone else’s code or show yours to someone else. Never be in possession of someone else’s code (except your partner).

One-on-One Sessions (optional)
Next 1.5 weeks, we are holding 1/2-hour one-on-one sessions on a computer with each student in CS1110.

Purpose: Help you develop a class as preparation for A1, give you a chance to ask questions. Not counted in final course grade.

Sign up on the course CMS (http://cms.csuglab.cornell.edu): Click on assignment One-on-one, find the schedule of times/instructors. Choose one.

Bring to the 1-on-1: the book, laptop w. DrJava if you have one.

Students with little or no programming experience report that these sessions are extremely helpful!

Office hours: [URL]
Already started: Prof Gries & Prof Lee (Tu/Th 10:10-10:55 Hollister 202), consultants TAs: will begin the week of Sept. 20, check the URL then

Field: a variable that is in each folder of a class.

```java
public class Worker {
    /** An instance is a worker in a certain organization. */
    private String lname; // Last name ("" if none; never null)
    private int ssn; // Social security #: in 0..999999999
    private Worker boss; // Immediate boss (null if none)

    Declarations of fields
    
    **/ An instance is a worker in a certain organization. */
    public class Worker {
        private String lname; // Last name ("" if none; never null)
        private int ssn; // Social security #: in 0..999999999
        private Worker boss; // Immediate boss (null if none)
    }

    Usually, fields are private, so methods that are outside the class can’t reference them.
    (Exception: private fields are accessible in the DrJava Interactions pane)
```

Initialize fields when a folder is first created

We would like to be able to use something like

```java
new Worker("Obama", 1, null)
```

to create a new Worker, set the last name to “Obama”, the SSN to 000000001, and the boss to null.

For this, we use a new kind of method, the constructor.

Purpose of a constructor: to initialize (some) fields of a newly created object

This initialization should make the class invariant true.

```java
public class Worker {
    private String lname; // Last name ("" if none; never null)
    private int ssn; // Social security #: in 0..999999999
    private Worker boss; // Immediate boss (null if none)

    Declarations of fields
    
    **/ An instance is a worker in a certain organization. */
    public class Worker {
        private String lname; // Last name ("" if none; never null)
        private int ssn; // Social security #: in 0..999999999
        private Worker boss; // Immediate boss (null if none)
    }

    Usually, fields are private, so methods that are outside the class can’t reference them.
    (Exception: private fields are accessible in the DrJava Interactions pane)
```
In the class definition of Worker:

```java
/** Constructor: an instance with last name n (can’t be null, can be ""), SSN s (an int in 0..999999999), and boss b (null if none). */
public Worker(
    String n,
    int s,
    Worker b)
```

Example constructor

```java
Worker lname = n;
ssn = s;
boss = b;
}
```