CS1110 8 Sep 2009 Customizing a class

Summary of lectures: On course home page, click on “Lectures” and then on “Outline of lectures held so far”.

Reading for this lecture: Sections 1.4, (p. 41); 13.3.1 (p. 376).

Read all “style notes” and referenced PLive lectures (activities).

Quote for the day: I have traveled the length and breadth of this country and talked with the best people, and I can assure you that data processing is a fad that won't last out the year.
—Editor in charge of business books for Prentice Hall, 1957

Reading for next lecture:
- Fields; getter & setter methods. Secs 1.4.1 (p. 45) & 3.1 (pp. 105–110 only)
- Constructors. Sec. 3.1.3 (p. 111–112)
- Testing. App. I.2.4 (p. 486)
One-on-One Sessions

Next two weeks, 1/2-hour one-on-one session on a computer with each student in CS1110

**Purpose**: See how well you understand what we have done, let you ask questions, give you help. Graded 0-1: 1 if you did a session. Not counted in course grade. Purpose: to help you.

**Instructors**: Gries, Lee, TAs, consultants.

**How to sign up**: Visit CMS. Click on assignment One-on-one. Choose from list of times/instructors. First-come-first-served.

Quiz on Thursday

What a type is: (p. 7 of text)

How to execute the assignment (p. 28, box on top of page).
People learn differently.

**Learning styles**

- **active versus reflective learners**
  - learn by doing vs. learn by reflection; groupie vs. loner
- **sensing versus intuitive learners**
  - practical/careful vs. fast/innovative
- **visual versus verbal learners**
  - pics, charts, films vs. words, explanations
- **sequential versus global learners**
  - logical, step-by-step, bottom-up vs. big-picture

Course outline webpage has link to website of Felder and Brent where you can read about this and take a self-scoring test to see your strengths/weaknesses
Patient

- name: “B. Clinton”
- address: “New York”
- owes: $250.00

This reviews what we did last time.

- x has value 6
- y has value c1
- y.getName() has the value “B. Clinton”
- y.deposit(250) will change the value of field owes to 0.

procedure call

function call
Class `javax.swing.JFrame`: an object is a window on your monitor.

```java
new JFrame()
```

Expression: create a new object of class `JFrame` and yield its name

This reviews what we did last time.
**Class definition**: The java construct that describes the format of a folder (instance, object) of the class.

```java
/** description of what the class is for */

public class <class-name> {

    declarations of methods (in any order)

}
```

This is a comment

A class definition goes in its own file named

```
<class-name> . java
```

On your hard drive, have a separate directory for each Java program that you write; put all the class definitions for the program in that directory.
**Class definition:** The java construct that describes the format of a folder (instance, object) of the class.

```java
/** description of what the class is for */

public class C extends <superclass-name> {
    // declarations of methods (in any order)
}
```

Class C has all the fields and methods that <superclass-name> does, in addition to those declared in C. Class C **inherits** the fields and methods of <superclass-name>.
/** description of what the class is for */

```java
public class subclass-name extends superclass-name {
    declarations of methods
}
```

**folder (object) belongs in file drawer for class**

**subclass-name**

methods and fields inherited from **superclass-name**

methods and fields declared in **subclass-name**
First example of a procedure and of a function

/** description of what the class is for */
public class subclass-name extends superclass-name {
    /** Set the height of the window to the width */
    public void setHeightToWidth() {
        setSize(getWidth(), getWidth());
    }
}

/** = the area of the window */
public int area() {
    return getWidth() * getHeight();
}
}
import javax.swing.*;
/** An instance is a JFrame with methods to square it and to provide the area of the JFrame */
public class SquareJFrame extends JFrame {
    declarations of methods
}

To the left, draw a manila folder of class SquareJFrame. When we define methods, put them in the proper place

folder (object) belongs in file drawer for class SquareJFrame
import javax.swing.*;

/** An instance is a JFrame with methods to square it and to provide the area of the JFrame */
public class SquareJFrame extends JFrame {
    /** = the area of the window */
    public int area() { … }

    /** Set the height equal to the width */
    public void setHeightToWidth() {…}
}

The class and every method in it has a comment of the form

    /** specification */

It is a Javadoc comment. Click on javadoc icon in DrJava to extract class specification. DO THIS AT LEAST ONCE IN LAB.
null denotes the absence of a name.

var3.getName() is a mistake! You get a NullPointerException