

CS1110 7 Sept 2010 Customizing a class

Summary of lectures: On www.cs.cornell.edu/courses/cs1110/2010fa, click on "Lecture summaries"

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 1.4.2 (p. 45) & 3.1 (pp. 105–110 only)
- Constructors. Sec. 3.1.3 (p. 110–112)
- ~~Testing. App. 1.2.4 (p. 486)~~

1

TakeNote®
 High quality lecture notes since 1985

IS AVAILABLE FOR THIS COURSE AT

www.tnote.com




FREE DELIVERY TO NORTH AND WEST CAMPUS

Questions? info@tnote.com

Informal Lunch with Gries or Lee

About once a week, perhaps a bit more.
 8 students max at a time. Sign up on the CMS.

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.

3

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

See 6 Sept NY Times article "forget what you know about good study habits: tinyurl.com/29f3vzx

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

4

QUIZ

1. Get out a blank piece of paper.
2. Write your LAST name, FIRST name, and Cornell NetID (not your Cornell ID. Gries's NetID is djg17)
3. Put two variables on the page, exactly like this:
 x y
4. Write down, in English, how to execute this assignment statement (don't explain how to evaluate the expression $x+y-1$):
 $y = x + y - 1;$
5. Execute this assignment statement, using the two variables you previously drew on your piece of paper.

5

This reviews what we did last time.

```

x  int
y  Patient
    
```

Patient class (type) of this object

- name "B. Clinton" (fields (they are variables))
- address "New York" (fields (they are variables))
- owes \$250.00 (fields (they are variables))
- getName() (assume is a function)
- deposit(double d) (assume is a procedure)

x has value 6
 y has value **c1** (function call)
 y.getName() has the value "B. Clinton"
 y.deposit(250); will change the value of field owes to 0.
 (procedure call)

6

Class java.awt.JFrame: an object is a window on your monitor

x A. It's all obvious, a piece of cake
 B. I'm okay, although I don't get it all
 C. I'm rather neutral. I can see it but I have to study more
 D. I'm very apprehensive about this course
 y E. I don't understand *anything*

```

    getX()  getY()  setLocation(int,int)
    getWidth() getHeight() setSize(int,int)
    ...
  
```

new JFrame()

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

This reviews what we did last time.

7

Class definition: The java construct that describes the format of a folder (instance, object) of the class.

```

    /** description of what the class is for
    */
    public class <class-name> {
        declarations of methods (in any order)
    }
  
```

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.

8

Class definition for a new subclass:

```

    /** 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>.

9

```

    /** description of what the class is for */
    public class subclass-name extends superclass-name {
        declarations of methods
    }
  
```

a0

superclass-name

methods and fields inherited from superclass-name

subclass-name

methods and fields declared in subclass-name

folder (object) belongs in file drawer for class subclass-name

10

First example of a procedure and of a function.

Note the specifications (comments) on the methods.

```

    /** 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();
        }
    }
  
```

11

```

    import javax.swing.*;
    /** An instance is a JFrame with methods to square it and
    An object of class java.util.Date contains the date and time at
    which it was created.
    It has a function toString(), which yields the date as a String.
    Write a procedure setTitleToDate, which will set the title of
    the window to the date.
    Body is:
    A. setTitle(new java.util.Date());
    B. setTitle(" " + new java.util.Date());
    C. setTitle(new java.util.Date().toString());
    D. None of these
    E. I don't know
  
```

12

```

import javax.swing.*;                                Javadoc
/** 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() { ... }

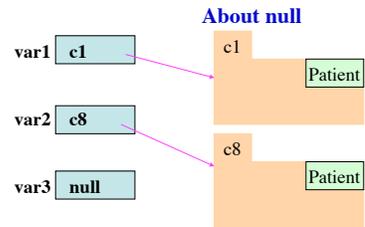
    /** Set the height equal to the width */
    public void setHeightToWidth() {...}
}

```

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**.