Slide 1

CS1130 Spring 2011: David Gries & Walker White

website: www.cs.cornell.edu/courses/cs1130/2012sp/

Transition to Object-Oriented Programming

Assumes programming knowledge in a language like Matlab, C, C++, or Fortran. Students who have learned Java but were not exposed heavily to OO are welcome.

Course Outcome: Understand OO concepts, as used in Java: classes, subclasses, inheritance, and overriding. This includes an operational model of method calls and the basics of OO design.

Lectures: M, F 2:30 or 3:35
Lab: W   2:30 or 3:35 or 7:00
(no need to attend but work must be done by end of the week)

1

Slide 2

1. Labs. Weekly, in Holl. B14. Bring laptop. TAs/consultants will help.
2. Quizzes. Let you know what material is important for you to know at that point. Quizzes require mastery: take them until you pass.
3. Course text: CD at the back of book has 250 2-4 minute lectures, each on 1 specific point. CD missing? See course website.
4. Two prog assignments require mastery. Submit, get feedback, resubmit, … until it is right.
5. Two tests require mastery.
6. Piazza. Our "town square" —ask and answer questions.
7. Consultants available for help in Green room of ACCEL lab. Visit course website to find schedule.

2

Slide 3

Academic Integrity. We ask you not to cheat, in any way, shape, or form. On our side, we try our best to be fair about the amount of work, in grading the work, and in giving you a course grade. For more info, see course website. **Do Quiz 0 on Course CMS.**

Course Management System. AFTER TUESDAY:

Visit cms.csuglab.cornell.edu/ CS1130lectures not listed? Email Maria Witlox, mwitlox@cs.cornell.edu, ask to add you to CS1130lectures . Include your Cornell netid in your email.

3

Slide 4

Four things to do

1. AFTER TUESDAY: Get on Course Management System (CMS). Get to it from link on course website.

Not registered on our CMS? Email Maria Witlox and ask her to register you.
mwitlox@cs.cornell.edu
She needs your netid.

2. Get DrJava onto your computer. See course website, Piazza for info. Problems with it? You can wait, but do try.

3. Academic Integrity. Read about it on course website. Visit CMS and do Quiz 0.

4. Read in course text (see next slide)

course website: www.cs.cornell.edu/courses/cs1110/2010fa/

4

Slide 5

Reading for this and the next lecture:
Sections 1.1, 1.2, 1.3. Lab 1: practice with concepts and details of 1.2, 1.3. You will not understand all the reading because there are many new terms, but doing the reading will enhance next lecture.

PLive: Lesson 0, Lesson page 1.3, Activity 1-4.1.

Summary of lectures: On course website, click "Lecture summaries".

Today Introduce types, expressions, variables in
         Show you around the CD **PLive**

DrJava. We write programs using the free IDE (Integrated Development Environment) called DrJava. Download it from the course website.

5

Slide 6

Type: A set of values together with operations on them.

Memorize definition! Write it down several times.

Type **integer**:

values: …, −3, −2, −1, 0, 1, 2, 3, 4, 5, …

operations: +, −, *, /, unary −

Use Chapter 6 as a reference for primitive types!

Type **int**: $-2^{31} .. 2^{31}-1$

values: −2147483648, −2147483647, …, −3, −2, −1,
       0, 1, 2, 3, 4, 5, …, 2147483646, 2147483647

operations: +, −, *, /, unary −

6

Strongly typed versus weakly typed

Matlab: weakly typed. A variable can contain a number at one point, and a string (of characters) at another point, and a 2-dimensional array at another point of execution.

Java: strongly typed. A variable has to be declared with a type before it is used, and it can contain values only of that type.

Variable: named box with a value in it.   b | 35 |

DrJava: Set a preference so that variables need not be declared in the interactions pane before they are used in the interactions pane. Menu item Edit -> Preferences; click on Interactions Pane.

Primitive types

byte	1 byte
short	2 bytes
int	4 bytes
long	8 bytes
float	4 bytes
double	8 bytes
char	2 bytes
boolean	2 bytes

7

Type: A set of values together with operations on them.

mantissa   exponent

Type **double**:
values: Examples: –22.51E6   equivalent to –22510000
or –22.51 * 10^6

22.51E–6   equivalent to .00002251
or   22.51 * 10^{-6}

An approximation to the real numbers.
operations: +, –, *, /, unary –

Type boolean

Values: **true   false**

Operators: and **&&**   or **||**   not **!**

Use Chapter 6 as a reference for primitive types!

8

Declaration of a variable

Examples   Syntax: <type> <variable> ;

int b;   b | 0 | **int**   b contains an **int**

double d;   d | 0.0 | **double**   d contains a **double**

String s;   s | a1 | String   s contains the name of a String object

int[] c;   c | a2 | **int**[]   c contains the name of a 1-dim array object

double[][] a; a | a9 | **int**[][]   a contains the name of a 2-dim array object

9

Assignment statement

Syntax: <variable> =   <expression> ;

Semantics: To execute the assignment, evaluate the <expression> and store its value in the <variable>.

b= b + 1;
d= d*5 + b;
d= b;

b= d; is illegal. Won't compile

Note:
<variable> must be as "wide" or "wider" than <expressions>

b | 0 | **int**

d | 0.0 | **double**

d is wider than b

10

Casting

(**int**) 3.5 is a cast;
it casts **double** value 3.5 to type **int**

b= (**int**) 3.5;

d= (**double**) 4;

d= 4;   4 has to be cast to **double**; it is a widening cast, so Java does it automatically.

b | 0 | **int**

d | 0.0 | **double**

d is wider than b

narrow   wide
byte -> **short** -> **int** -> **long** -> **float** -> **double**
char -> **int** -> **long** -> **float** -> **double**

11

Precedence of operators (page 23)

- Unary operators: + – !
- Binary arithmetic: * / %
- Binary arithmetic: + –
- Arithmetic relations: < > <= >=
- Equality relations: == !=
- Logical and: &&
- Logical or: ||

The next lecture will also discuss:
Types boolean (p. 20) and String (p. 22)
You will use these things in Lab 01.

12