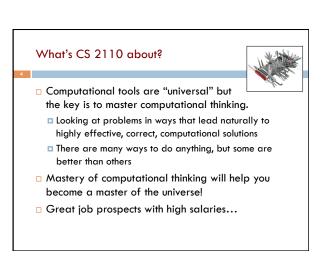


Welcome to CS2110! 054 Freshmen Instructors: 220 Sophomores Nate Foster, David Gries, 045 Juniors Recitation leaders: 18 053 Seniors 076 Meng Consultants: 25 034 Graduate Above numbers were from several days ago. Last look at enrollment: 478.

Velcome to CS2110! Learning about: OO, abstract data types, generics, Java Collections, ... Reasoning about complex problems, analyzing algorithms we create to solve them, and implementing algorithms with elegant, easy-to-understand, correct code Testing; Reasoning about correctness Data structures: linked lists, trees, graphs, etc. Recursion Algorithmic complexity Parallelism —threads of execution



Is CS2110 right for you? | Knowledge of Java not required | Only ~30% of you know Java –others know Matlab, Python ... | Requirement: comfort with some programming language, on the level of CS1110 (Python based) and CS1112 (Matlab based). | Prior knowledge of OO not required. | We assume you do not know Java!

Homework 1. Read article Why Software is So Bad. Link: Course website -> Lectures notes (Lecture 1) Homework 2. Get Java and Eclipse on your computer Homework 3. Spend some time perusing the course website. Look at course information, resources, links, etc. Homework 4. BEFORE EACH LECTURE: download the pdf form of the slides, bring them to class, and look through them during the lecture. We will be projecting not only PPT but also Eclipse and other things; having the PPT slides in paper form or on your laptop/tablet can help you during the lecture

Homework!

Lectures

- □ TR 10:10-11am, Statler auditorium 🏁
 - Attendance mandatory



- Same course! We call it CS 2110 in online materials
- □ Non-engineers sign up for CS 2110
- □ Engineers sign up for ENGRD 2110

Sections (Recitations)

T 12:20 4 sections: 44, 44, 21, 0

T 1:25 2 sections: 46, 25 T 2:30 2 sections: 40, 4

T 3:35 1 section: 27

W 12:20 2 sections: 36, 22 W 01:25 2 sections: 33, 18

W 02:30 1 section: 27, 6 W 07:30 1 section: 10 Some time EARLY, visit StudentCenter and change your section to even out the numbers

Attendance is mandatory

Sometimes review, help on homework, new material

No permission needed to switch sections, but do register for whichever one you attend

CS2111

- □ An "enrichment" course
- We want to help students who might otherwise feel overwhelmed by CS2110
- Gives more explanation of core ideas behind Java, programming, data structures, assignments, etc.
- □ Taught by Foster and Gries, 1 credit S/U
- Only for students who also take CS2110
- □ Only requirement: Attend weekly lecture

Academic Excellence Workshops

- □ Two-hour labs: students work together in cooperative setting
- □ One credit S/U course based on attendance
- □ Time and location TBA
- □ See website for more info:

www.engineering.cornell.edu/academics/undergraduate/curriculum/courses/workshops/index.cfm

Piazza

- □ Click link on our web page to register
- □ Incredible resource for 24 x 7 help with anything
- We keep an eye on it and answer questions.
 YOU can (and will)
 too. Visit the Piazza often.



Resources

- Book: Frank M. Carrano, Data Structures and Abstractions with Java, 3nd ed., Prentice Hall
 - 2nd edition is okay. E-book not required
 - □ Share textbook. Need access to it from time to time
 - □ Copies on reserve in Engr Library
- PPT slides (on course website and Piazza) outline all of OO in Java. Has index at beginning
- Great Java resource: online materials at Oracle JDK web site.
 Google has it indexed.
- VideoNote: videos of lectures from Spring 2014.
 http://www.videonote.com/cornell. Log in with netid

Obtaining Java



- □ Follow instructions on our Resources web page
 - Make sure you have Java JDK 1.8, if not download and install. We explain how on the web page.
 - □ Then download and install the Eclipse IDE
- □ Test it out: launch Eclipse and click "new>Java Project"
 - □ This is one of a few ways Java can be used
 - When program runs, output is visible in a little console window

Eclipse IDE



- □ IDE: Integrated Development Environment
 - □ Helps you write your code
 - □ Protects against many common mistakes
 - □ At runtime, helps with debugging
- □ Follow Resources link to download and install



DrJava IDE



- □ IDE: Integrated Development Environment
- □ DrJava is a much simpler IDE, few features
- We use it only to demo Java features and programming concepts. Has an "interactions pane", which allows trying things without requiring a complete Java program.
- □ DON'T use it for course assignments –use Eclipse
- □ Free at www.drjava.org

Coursework

16

- 7–8 assignments involving both programming and written answers (35%)
- □ Two prelims (15% each)
- □ Final exam (30%)
- Course evaluation (1%)
- □ Possible surprise in-class quizzes (4%)

Formula will change as the course progresses and we make changes in assignments, give quizzes, etc.

Exams are most important aspect in determining final grade

Assignments



Teams of one or two

- $\hfill \Box$ A0 and then A1 will be posted soon on the CMS
- Finding a partner: choose your own or contact your TA. Piazza can be helpful.

Two kinds of assignment:

Vanilla: specific experience to learn and practice what's being taught. We give exact instructions for doing it

Chocolate: Open-ended project done in 3 chunks Parts of the design are left to you.

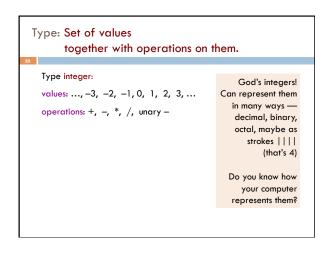
CS 2111 will give more help on it.

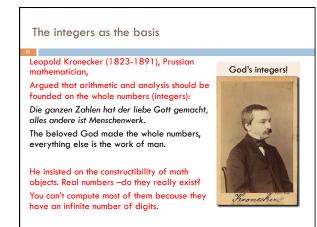
Academic Integrity... Trust but verify!

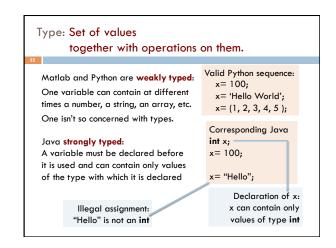


- $\hfill 98\%$ of you are honest and don't try to cheat
- □ We use artificial intelligence tools to check each homework assignment, so catch the other 2%
 - ☐ The software is very accurate!
 - □ It tests your code and also notices similarities between code written by different people
- □ Sure, you can fool this software
 - ... but it's easier to just do the assignments
 - ... and if you try to fool it and screw up, you might fail the assignment or even the whole course.

References in text and in JavaSummary type: A.14 slide 4 variable: A.13 slide 7 variable declaration: A.15 slide 7 Primitive types, A.16, back inside cover slide 5 Constants, A.17 slide 20 Assignment, A.18-A.20 slide 8 Casting, A.21 slide 6 Expressions: A.22-A.23 Precedences: A.24, back inside cover Unicode character codes, back inside cover





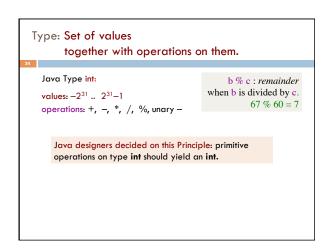


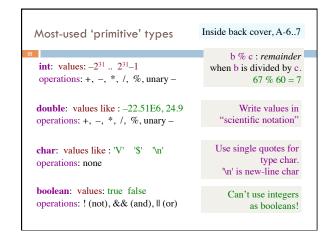
Weakly typed:

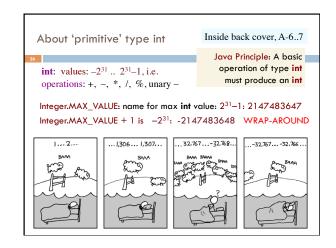
Weakly typed:
Shorter programs, generally.
Programmer has more freedom, language is more liberal in applying operations to values.

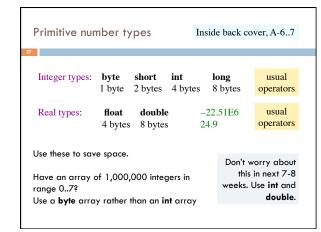
Strongly typed:
Programmer has to be more disciplined. Declarations provide a place for comments about variables.
More errors caught at compile-time (e.g. it's a syntax error to assign a string to an int variable).

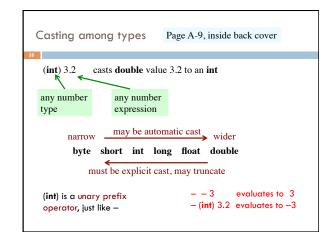
Note: weak and strong typing not well defined; literature has several definitions



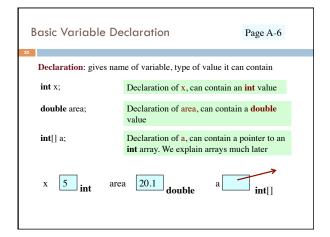


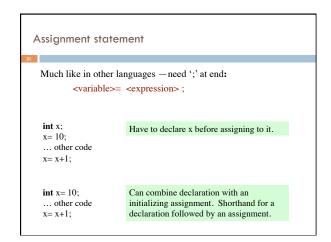


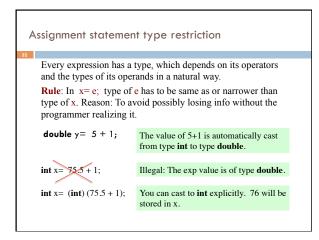




Char is a number type! Page A-9, inside back cover char is a number type: (char) 86 'V'Unicode repr. in decimal: 86 Unicode: 16-bit char repr. Encodes chars in just about all languages. In java, use hexadecimal (base 16) char literals: '\u0041' is 'A' '\u0950' is 'ॐ' '\u5927' is '大' '\u885b' is '衛' -Om, the sound of the universe '\u0042' is 'B' —大衛 is (I think) a transliteration of David into Chinese (Da Wei) '\u0024' is '\$' See www.unicode.org







```
A function in Matlab, Python, and Java
  function s = sum(a, b)
     % Return sum of a and b
  s=a+b;
  def sum(a, b):
      """ return sum of a and b"""
      \textbf{return}\ a+b
                                                       Specification:
                                                          in comment
  /** return sum of a and b */
                                                     before function
  \textbf{public static double} \ sum(\textbf{double } a, \textbf{double } b) \ \{
     return a + b;
                                                     Declarations of
                                                parameters a and b
              return type
```