

Welcome 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

5

- Algorithmic complexity
- Parallelism —threads of execution

Homework! 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

What's CS 2110 about?



- Computational tools are "universal" but the key is to master computational thinking.
 - Looking at problems in ways that lead naturally to highly effective, correct, computational solutions
 - There are many ways to do anything, but some are far better than others
- Mastery of computational thinking will help you become a master of the universe!
- Great job prospects with high salaries...

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!

Lectures

TR 10:10-11 am, Statler auditorium ^{And}
 Attendance mandatory



- □ ENGRD 2110 or CS 2110?
 - **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)

- □ Like lecture, attendance is mandatory
- □ Sometimes review, help on homework
- Sometimes new material
- □ Section numbers are different for CS and ENGRD
- Each section led by member of teaching staff
- No permission needed to switch sections, but do register for whichever one you attend

T 12:20 3 sections: T 1:25 3 sections: T 2:30 2 sections: T 3:35 1 section:	32, 34. 16 30, 12	Some time BEFORE Tuesday, visit StudentCenter and change your section to even out the numbers
W 12:20 3 sections: W 01:25 2 sections: W 02:30 1 section:	34, 30, 18 33, 33, 19	
	16	

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 Gries and James, 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

11

□ See website for more info:

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

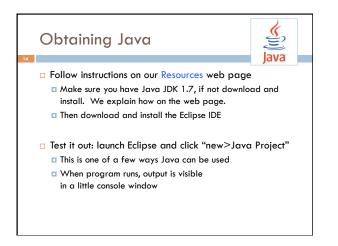
Piazza

- 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
 - $\hfill\square$ 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



eclipse **Eclipse IDE J**drjava DrJava IDE 15 IDE: Integrated Development Environment IDE: Integrated Development Environment Helps you write your code DrJava is a much simpler IDE, few features Protects against many common mistakes □ We use it only to demo Java features and At runtime, helps with debugging programming concepts. Has an "interactions pane", □ Follow Resources link to download which allows trying things without requiring a and install complete Java program. DON'T use it for course assignments -use Eclipse □ Free at www.drjava.org

Coursework

17

- 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

- 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!

19



 We use artificial intelligence tools to check each homework assignment

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.

Types in Java 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

Type: Set of values together with operations on them.

The integers as the basis

Leopold Kronecker (1823-1891), Prussian mathematician,

Argued that arithmetic and analysis should be founded on the whole numbers (integers):

Die ganzen Zahlen hat der liebe Gott gemacht, alles andere ist Menschenwerk.

The beloved God made the whole numbers, everything else is the work of man.

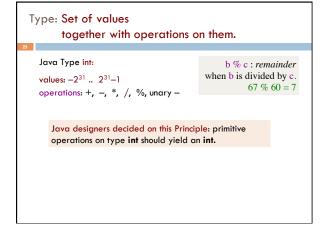
He insisted on the constructibility of math objects. Real numbers –do they really exist? You can't compute most of them because they have an infinite number of digits.

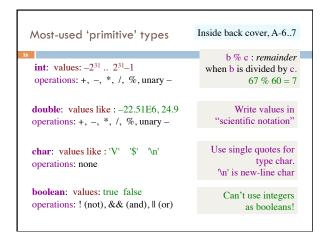


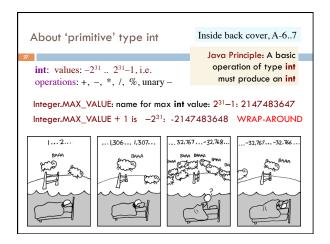
God's integers!

Туре	: Set of values together with operations	on them.
One times	ab and Python are weakly typed : variable can contain at different a number, a string, an array, etc. isn't so concerned with types.	Valid Python sequence: x = 100; x = 'Hello World'; x = (1, 2, 3, 4, 5);
Java A vai it is u	strongly typed: iable must be declared before sed and can contain only values e type with which it is declared	Corresponding Java int x; x= 100; x= "Hello";
	lllegal assignment: "Hello" is not an int	Declaration of x: x can contain only values of type int

Weakly typed versus strongly 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







Primitive nur	nber tγ	/pes		Inside	back co	ver, A-67
Integer types:	byte 1 byte	short 2 bytes			ng bytes	usual operators
Real types:	float 4 bytes	double 8 byte	-	-22.5 24.9	51E6	usual operators
Use these to sav Have an array o range 07? Use a byte arra	of 1,000,		-		this	orry about in next 7-8 Jse int and double .

