

Lecture 01:

Welcome and Logistics

Sierra College

CSCI-12

Spring 2015

Mon 01/26/15

WELCOME

TO

CS-12

Today's Topics

- Scope of this course
- Goals of this course
- Course Overview
- Roster/Adds
- Instructor
- Syllabus
- Canvas
- Advice from past students
- Advice from instructor
- Your “to-do” list
- Add codes at end

Scope Of This Course

- Introductory concepts of programming course
 - First “real” programming course in this department
 - Java language, object-oriented (O-O) principles
 - Today: a bit about Java, much more over coming lectures
- 16 weeks, 3.0 units
 - We’ll cover most of Chs.1-8, along with some other material
 - Somewhat in order, but I will vary the order of the material
 - Some topics omitted due to time constraints, covered in CS-13
- Past programming experience is NOT required
 - Newcomers: we will be starting from scratch and first principles
 - Experienced programmers: we will be talking about Java in common, global, “language-agnostic” terms
 - Who has past programming experience? Which language(s)?

Goals Of This Course

- Introduction to programming concepts, using Java
- Learn how to first use, and later create, reusable software components called **objects**
- Learn to write well-structured, well-documented software, using sound software development practices
- Decompose Java into standard, universal elements of programming, which are common and transferable across languages

The “Big Picture” For This Course

Classes: blueprints for “things”
(software components)



Objects: individual “things”
created from the blueprint



Roster/Adds

- Roster
 - Need to do a “roll call” to see who is/isn’t here today
 - You must be in attendance at the first class meeting
 - Class roster is as of earlier this morning
 - This also give me the opportunity to start putting names to faces!
- Adds
 - Limited number of add codes at **end** of lecture, dependent upon seating and reasonable class size
 - CS-10 is now an enforced, formal prerequisite for this course

Instructor Bio

- **5th semester teaching this course at Sierra College; adjunct faculty**
- **~28 years working in various aspects of software**
 - Software design/development, application support, contract consulting, QA/testing, writing app notes/examples, training/teaching, teaching assistant
 - Aerospace, design automation software, modeling & simulation, robotics
 - Silicon Valley, Davis/Sacramento, San Diego
 - University (UCD), community college (Sierra), and professional education (UC Extension) levels
- **Original academic background**
 - Mechanical Engineering (B.S. & M.S., UC Davis)
 - 5 years in industry in-between degrees
 - So... my focus is on the practical, problem-solving aspects of software
- **2 A.S. degrees and the Web Programming certificate from Sierra College**
 - Completed most of the CS courses offered here
 - Took classes from many of the current CS profs
 - Sat exactly where you are sitting right now
- **Programming in 12-14 languages**
 - The first one was tough, but later ones easier and easier
 - I see the common aspects of languages, which motivates my approach to this course
- **Had a rough first programming experience**
 - 30+ years ago at UCD (Fortran 77): I barely passed MY first programming class (retook it later for an A)
 - Gives me lots of empathy for other learners
 - Motivates me to make software understandable to others

Syllabus

- **Syllabus → The “User’s Manual” for this class**
- At A Glance
- Course Description
- Course Outcomes
- Schedule
- Communications
- Course Materials
- Course Structure
- Grading Policies
- Enrollment Policies
- Student Rights and Responsibilities
- Classroom Etiquette
- Academic Honesty
- Guiding Philosophies
- Suggestions for Success
- Emergency Procedures

Academic Honesty

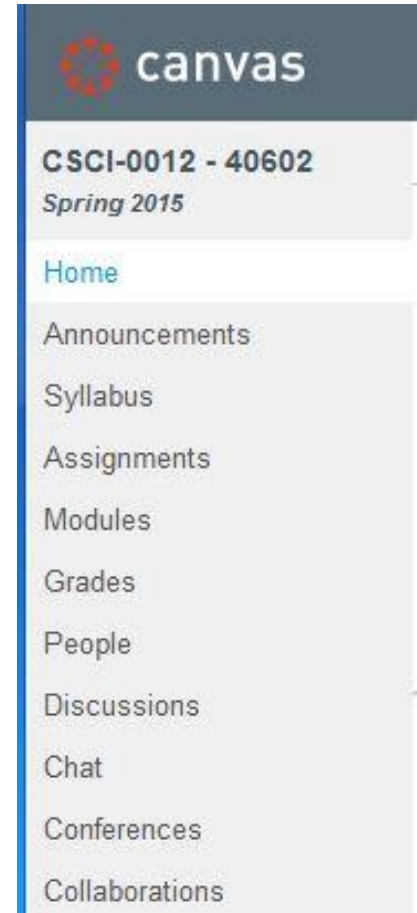
- We will review the syllabus statement on academic honesty specifically.
- Mutual discussions about assignments are OK, but ALL work submitted must be your OWN work.
No exceptions!
- Don't cheat or shortcut your own understanding of the course material.
- Plenty of help will be available to you to learn this material, please seek it out.

Canvas

- **Canvas** has been Sierra's campus-wide Learning Management System (**LMS**) since Spring 2013
- Canvas will be our go-to source for EVERYTHING concerning this course
 - Schedule, syllabus
 - Lecture notes
 - Code examples, links to resources
 - Assignments
 - Grades, feedback
 - Communications with instructor, other students
- EVERYTHING needed for your success in this course is in Canvas... so learn your way around it

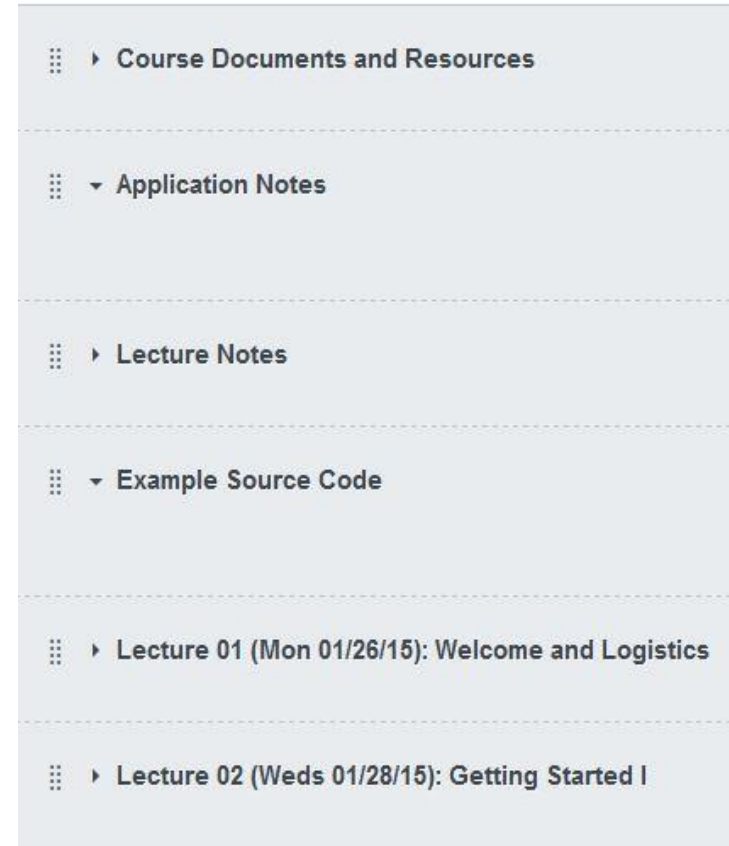
Canvas Navigation

- Left-side navigation menu is shown
- **Home**
 - An overview of what's where in Canvas
- **Announcements**
 - Course-wide messages from instructor, typically weekly
- **Syllabus**
 - Links to syllabus and schedule, as well as a list of all assignments
- **Assignments**
 - Details for all assigned work
- **Modules**
 - **Most of the course “substance” will be in here (next slide)**
- **Grades**
 - All your grades, weighted by Assignments/Exams
- **People**
 - Your course colleagues
- **Discussions**
 - Introductions is first one from me
 - Students can initiate these as well, ask questions of each other, etc.
- **Chat, Conferences, Collaborations**
 - Canvas-provided communication tools



Canvas Modules

- Canvas modules list is shown at right
- **Course Documents and Resources**
 - Syllabus, schedule, study guides, coding standard
 - Links to useful outside Java resources
 - Also, some Sierra College academic resources
- **Application Notes**
 - More detailed notes on specific topics (PDF)
- **Lecture Notes**
 - Instructor's lecture slides (PDF)
 - **I will provide ALL my lecture notes**
- **Example Source Code**
 - Instructor-created source code examples (.java)
- **Lecture NN**
 - All resources for a specific date's lecture



Advice From Your CS-12 Predecessors

- Final question on my Fall 2014 final exam:

Please write some short “lessons learned” to next semester’s CS-12 students, regarding how to succeed in this course, things you think they’d need to know, or anything else about the course that comes to mind.

Unless you indicate otherwise, I’d like to share your (anonymous) comments with them at the beginning of next semester, just as I did with you back in August.

- I transcribed all their responses verbatim
- Then, I grouped their responses into broad, general topics
- The following comments are their raw, unedited words

Fall 2014: On Lecture...

- *“Download the PowerPoint and have them handy; they are a wonderful reference (along with your own code).”*

Fall 2014: On Time Management...

- *“Stay on top of your work, and try not to break your keyboard!”*
- *“Great class. Don’t put off assignments till the last second. They get longer as you go.”*
- *“Start writing the classes ASAP - can take a quick worker 7+ hours to complete.”*
- *“Don't wait to start on the longer assignments, you'll run into problems, where you'll need help or to look at it with fresh eyes the next day.”*

Fall 2014: On Time Management...

- *“Keep up with the projects as well as you can so that you can use your own code as notes (dual purpose!)”*
- *“Don't fall behind, ask questions, look at all documentation professor gives you, don't be afraid to use Google - there are a TON of resources online. This class can get tough, but it is completely manageable if you put in the time.”*
- *“Programming is an acquired skill that you always improve at. Sometimes it is hard, but keep at it and eventually you will get it. Know when to walk away and try on a later day, it can be frustrating, at times.”*

Fall 2014: On Experimenting...

- *“Take time to do extracurricular experimentation in your IDE, you will learn traps and pitfalls, as well as understand the language better than can be learned from just this class.”*
- *“Reading the lectures and asking questions can only get you so far, you need to practice with the code, or you won't understand how it works, or how to write it.”*

Fall 2014: On Coding Techniques...

- *“Get in the habit of drawing pictures, outlines, or anything to help visualize the logic/code you want to achieve. Coding is a problem solving skill that you won't always be able to visualize in your head.”*
- *“Build your own skeleton of braces BEFORE writing code in it. Hunting for the extra braces/parens is very annoying.”*

One Final Thought

“There is no shame in not knowing...

... the shame lies in not finding out.”

- Russian proverb

Things To Do For Next Time

- Review the syllabus
- Get familiar with the layout of this course in Canvas
- Obtain your textbook
- Finalize your enrollment(s), if applicable
- Next time, we will have a “getting started” lab
 - A “quick-and-dirty” intro to Java and our development tool
 - I’ll talk thru/demonstrate the lab exercise in lecture
 - You’ll implement the same program in lab
 - **Make sure you have some means of file storage (USB drive or cloud storage account) by next time!**
- We will also have a Canvas intro assignment (3 short parts)
 - Post a Canvas discussion board intro
 - Canvas messaging/syllabus
 - VARK inventory (learning styles)