

# CSCI 1103: Course Mechanics

Chris Kauffman

*Last Updated:  
Fri Sep 8 16:56:40 CDT 2017*

# Expectations

## Kauffman can

- ▶ Provide guidance, entertainment, information, challenge
- ▶ Will do all of those in lecture, office hours, projects, exams

## Kauffman cannot

- ▶ Force you to pay attention, do your HW, attend labs, learn
- ▶ Cannot force you to **care**, the most important aspect in CS or any education

## Kauffman's Expectation

- ▶ You care some and will cultivate a further attitude of curiosity and discipline
- ▶ You will put some effort into our time together as I have

# Overview of Mechanics: Syllabus has Details

## Lecture

- ▶ Thrice per week, 50 minutes
- ▶ Do what we did today: talk, grill, code, laugh
- ▶ 2 exams and a final

## Projects

- ▶ 6 planned, **individual work**
- ▶ Larger than labs, several weeks
- ▶ Will discuss in lectures

## Canvas:

<http://canvas.umn.edu>

- ▶ All important links like syllabus, schedule, specs, slides
- ▶ Submit assignments, get grades

## Monday Lab

- ▶ Mondays, ~2hrs in a computer lab
- ▶ Do programming exercises, Mandatory Attendance
- ▶ Work with a partner: limited computers
- ▶ **Free collaboration** on labs with anyone in our course
- ▶ Due by end of week, must submit yourself

## Piazza

- ▶ Discussion board
- ▶ Staff will answer questions
- ▶ Read Etiquette Post

# Lab Policies

## Attendance

- ▶ You may only attend the lab section for which you are registered. If that is a problem, see Prof. Kauffman during office hours.
- ▶ Attendance at the first lab meeting is mandatory.
- ▶ Attendance at labs after the first meeting is optional.

## Partners

- ▶ You may work in groups of 2 but BOTH members
- ▶ Must submit code to Canvas
- ▶ Must be present for the Check-off
- ▶ You may freely collaborate with anyone in 1103 on labs but obey the PRIME DIRECTIVE (syllabus)

## Grading: 0-100%

- ▶ Check-off 40%: demonstrate to a TA that your lab code works This must be done in person during lab or during TA office hours.
- ▶ Submit 60%: submit required files according to the lab instruction. This can be done without attending.

## First Week Assignment: Agree to Syllabus

All students must submit an agreement to abide by the syllabus.

- ▶ Shows you can edit a text file
- ▶ Can submit assignments to the course Canvas site
- ▶ Download the text document here:

<http://cs.umn.edu/~kauffman/1103/agreement.txt>

I have familiarized myself with the contents of the CSCI 1103 syllabus and agree to abide by the policies contained within it. I will obey the PRIME DIRECTIVE. As a University of Minnesota student, I agree to follow the Student Code of Conduct and will treat my classmates and the course staff with honest respect.

Signed,  
(YOUR NAME HERE)

Fill in your name with your favorite text editor:

- ▶ Windows Notepad
- ▶ Mac TextEdit
- ▶ DrJava: Course program editor

# We're on the Web

## Piazza: Discussion Board

- ▶ Project and Lab discussion
- ▶ Questions about programming
- ▶ Announcements from Staff
- ▶ Course Schedule
- ▶ 95% of communication/questions
- ▶ Read the etiquette post (up shortly if it's not already)

## Email for

- ▶ Appointments outside of office hours
- ▶ Unresolved grading disputes
- ▶ Personal emergencies/problems

## Canvas for

- ▶ Assignment Submission
- ▶ Grades

# Lecture

## Mechanics

- ▶ Talk
- ▶ Code
- ▶ Try
- ▶ Ask

## Hot Seats

- ▶ Front few rows are **hot seats**
- ▶ I will grill hot seats
- ▶ Just try: answer questions, give feedback
- ▶ Up to 3% overall bonus
  - ▶ Susy has 20 pts, max in class, 3% bonus
  - ▶ Sammy has 10 pts, 1.5% bonus
- ▶ Don't want/need participation, don't sit in hot seats
- ▶ Don't like lectures, don't come, but don't complain if you miss something
- ▶ *Someone* is paying \$1969.24 or more for the privilege of you being in this room (4-credits, in-state tuition)

# Reading

## Introduction to Programming Using Java, Seventh Edition

- ▶ Author: David J. Eck
- ▶ Freely available: <http://math.hws.edu/javanotes/>
- ▶ Presumes no programming background
- ▶ Decent and cheap
- ▶ *Lots* of Java books/tutorials out there feel free to explore and share good links
- ▶ *Alternative: Building Building Java Programs (Reges/Stepp)*

## Java Docs

- ▶ Official documentation of Java library
- ▶ Becomes more important later in the class



# Making Programs

Edit, Compile, Run, Fix: You need

- ▶ Text Editor (`jedit`, `emacs`, `vim`, `notepad`, etc)
- ▶ Compiler (`javac`)
- ▶ Run environment (command line like `cmd.exe` on mingw or `Terminal.app`)

That's it, the rest is gravy

# Making Programming Faster

- ▶ An IDE combines these things in a sensible way
  - ▶ Text Editor
  - ▶ Compiler interface button
  - ▶ Run interface
- ▶ Fanciness
  - ▶ Debugger interface
  - ▶ Testing interface
  - ▶ Documentation generation
  - ▶ File browser
  - ▶ Read, Eval, Print Loop for interactive testing

DrJava does all this, but...

All IDEs dumb down the act of creating programs and disguise many details.

# Tools

The official java tools of the course are

- ▶ [jdk 1.8](#), official build and run tools from Oracle
- ▶ [DrJava](#), a simple, superior java IDE (if you're into IDEs)
  - ▶ Download GMU edition:  
<https://cs.gmu.edu/~kauffman/drjava/>

Special Note:

- ▶ *I do not know how to use eclipse / netbeans / codesmacker*
- ▶ *I will not be learning how this semester.*
- ▶ *If I can help it I will never learn eclipse.*
- ▶ *TAs may be able to help you but are **not** required to do so.*
- ▶ *In class I will use DrJava, Emacs, and command line.*
- ▶ *If you have questions on those I'm happy to help.*

# Cheating

## Don't cheat

- ▶ Easy to catch
- ▶ Likely to get caught
- ▶ Painful for everyone (makes me ornery)
- ▶ You can't lie to nature

*For a successful technology, reality must take precedence over public relations, for Nature cannot be fooled.*

*– Richard Feynman, Challenger Disaster Report*

## Unsure if something constitutes cheating?

- ▶ **Stop and ask me**
- ▶ Sharing on Lab Exercises is fine
- ▶ Sharing on projects is not

# Cheating

**PRIME DIRECTIVE:** Be able to explain your own work including homework code and exam solutions. The work you submit should be the product of your own effort and reflect your personal understanding.

Follow this because...

*... I can say that at my workplace I've seen more than one freshout who clearly hadn't made it through college without significant assistance from Stack Overflow and other people's blogs. None of them lasted very long. Perhaps knowing how to solve problems for yourself isn't necessary to get a college degree nowadays, but it's surprising how useful it can be in a career where you solve problems for a living.*

– bunderbunder, Discussion of cheating using StackOverflow on <http://news.ycombinator.com/item?id=4910406>