CSE 331
Software Design & Implementation

Kevin Zatloukal

Fall 2020

Lecture 1 – Introduction & Administrivia

(Based on slides by Mike Ernst, Dan Grossman, and many others)

Administrivia

Who: Course staff

- **Instructor**: Kevin Zatloukal (kevinz at cs)
  - 15 years in industry, 5th year teaching
- 11 great **TA**s
  - mostly veterans
- Office hours posted soon
  - (starting later this week)

Get to know us!

  - We're here to help you succeed

Who: Students

- Assuming you have mastered CSE142 and CSE143
- Hoping (but not assuming) have you taken 311
  - will connect to 311 material where it arises
- Assuming you are in your second year of CS courses
  - seniors may be bored

Prerequisites

- Knowing Java is a prerequisite

Examples:

- Difference between **`int`** and **`Integer`**
- Distinction between **`x == y`** and **`x.equals(y)`**
- Aliasing: multiple references to the same object, what does assignment (**`x=y;`**) *really* mean?
- Subtyping via **`extends`** (classes) and **`implements`** (interfaces)
- Method calls: inheritance and overriding; dynamic dispatch
- Difference between compile-time and run-time type

Staying in touch

- Ed message board (link on course web page)
  - should have received an invitation already
  - best place to ask questions

- Course staff: **`cse331-staff@cs.washington.edu`**
  - For things that don't make sense to post on message board

- Course email list: **`cse331a_au20@u.washington.edu`**
  - students already subscribed (your UW email address)
  - frequent emails from me: one for each lecture
    - should show up at least 24 hours before lecture date

Lectures (part one)

- Includes pre-recorded videos (split into parts)
  - gave both live and recorded in Spring; latter was preferred
  - emailed *at least 20 hours before* the calendar date for topic

- Total lecture time will average **more than 50 minutes**
  - some longer and some shorter
  - required reading was reduced to compensate
    - feel free to watch at 1.25x speed
  - my bias is toward more teaching & learning, not less

- Will be editing & improving the recordings from Spring
  - "Introduction" lecture was mainly "spring me"
  - hoping editing of "fall me" with "spring me" is not too obvious

Lectures (part two)

- Includes a live Q&A session at 10:30 am on MWF
  - link to watch is on Canvas (see Zoom app)
  - these will also be recorded and posted on Canvas

- If you don't have questions, I encourage you to watch recording
  - can be hard to ask questions in front of 100 peers
  - others can get the questions and answers from the recording
  - exception: feel free to stop by to say "hi" this Wednesday

- Fine to ask questions about earlier lectures
  - (e.g., if you fall behind by a lecture)
  - but please watch the past Q&A recording and make sure that question was not asked and answered already

Section

- Will be focused on **helping with homework**
  - held on day HW is released
  - get you get you started with the work to be done
  - they should be very useful

- Live via Zoom video
  - links on Canvas (see Zoom app)

Homework Assignments

- Roughly 1 assignment per week

- First 3 are paper assignments
  - submit these in Gradescope
  - should get an invite email on Wednesday
    - let me know if you don't

- Remaining 7 are coding assignments
  - generally due on Thursday by 11pm
  - submit and tag your code in Gitlab
    - TAs will grade and get feedback to you

Homework Assignments

- Biggest misconception (?) about CSE331

  "Homework was programming projects that seemed disconnected from lecture"

- If you think so, you are making them harder!
  - approaching them as CSE143 homework won't work well
  - each HW designed to teach topics from prior lectures
  - seek out the connections by before typing

- (Tip: this is also true of quizzes)

Late Policy: Written Assignments

- Allowed only in special situations
  - let us know at least 30 hours beforehand
    - do not start the night before
  - will make exceptions for emergencies

Late Policy: Coding Assignments

- Same special situations as written assignments

- And also:
  - Up to **4** times this quarter you can turn in a homework assignment **one** day late
  - Not accepted for credit after that.
  - Late days are 24-hour chunks

- Why?
  - keep you on schedule (real world has deadlines)
  - get feedback to you before next deadline

Resubmission: Coding Assignments

- We will allow re-submission of coding assignments
  - send email to the staff (cse331-staff at cs)

- Aim of the policy is to limit the deductions for minor mistakes that end up causing a disproportionate number of test failures

- We will re-calculate the correctness score up to a maximum score of 80%
  - other scores (design, style, etc.) are not changed

Academic Integrity

- "The code you submit must be your own"
  - no copying from other students, web pages, etc.
- Read the full course policy carefully
  - ask questions if you are unsure
- Always explain in your HW any unconventional action
  - worst result then is some points lost
  - worst result otherwise is expulsion
- Violations are unfair to other students and yourself

Exams

- No regular exams this quarter
  - no midterm, no final

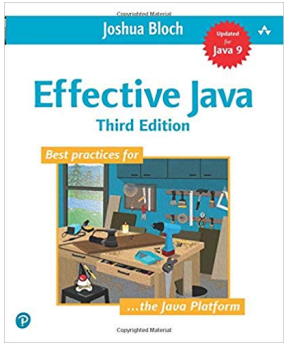
- Will have ≈7 **quizzes** during the quarter
  - 30-45 minutes each
  - mainly multiple choice
  - will have 24 hours in which to complete them
  - each test can be taken twice, with higher score used

- Will ask you to **demo** your final project to a TA
  - may ask you to explain how some parts are implemented, why you implemented them that way, etc.
  - all spring students got a 100% score

Books

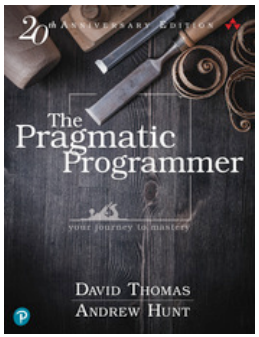
Required book

- *Effective Java* 3rd ed, Bloch (EJ)



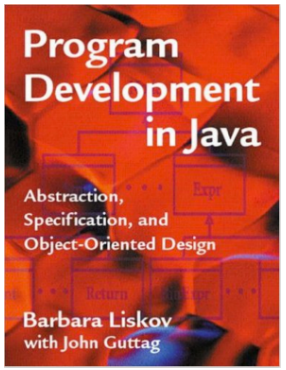
Optional book

- *Pragmatic Programmer*, new 20th anniversary (2nd) edition, Hunt & Thomas (PP)



Other books

- *Program Development in Java*, Liskov & Guttag
  - would be the textbook if not from 2001
- *Core Java* Vol I, Horstmann
  - good reference on language & libraries



Books? In the 21st century?

- Why not just use Google, Stack Overflow, Reddit, Quora, …?
- Web-search good for
  - Finding the parameters of a Java API function
- (can be) Bad for
  - Why does it work this way?
  - What is the intended use?
  - How does my issue fit into the bigger picture?

- Beware:
  - Answers on the web are often **quickly** out of date
    - aim is to answer the question at the time when asked
  - "This incantation solved my problem"
    - give that to users without knowing how it works?

Readings

- Calendar will include book sections for you to read
  - EJ = required, PP = optional
- These are “real” books about software, approachable in 331
  - occasionally slight reach: accept the challenge
- Overlap only partially with lectures
  - books include lots of other useful information
- Readings are fair game for quizzes
  - want to make sure you do it

Grading

Approximate weighting (subject to change):

65%	Homework
25%	Quizzes
10%	Final Project Demo

CSE 331 can be challenging

- Past experience tells us CSE 331 is **hard**
  - not my intention to make it difficult!
- Big change to move
  - **from** programming by trial & error
    - technique that does not work for building large scale software
  - **to** programming by careful design, reasoning, and testing
- Programming itself can be hard
  - surprisingly difficult to specify, design, implement, test, debug, and maintain even a simple program

CSE 331 can be challenging

- We strive to create assignments that are reasonable if you apply the techniques taught in class…
  … but likely hard to do in a trial & error manner
    … and almost certainly impossible to finish if you put them off until a few days before they're due

- Assignments will take more time than you think (**start early**)
  - even professionals *routinely* underestimate by 3x
  - these assignments will be a step up in difficulty

- If you are having trouble, *think* before you act
  - then, look for help

Other Advice

- Don't be afraid to make mistakes
  - accepting that you will make mistakes is perhaps the most important lesson of this course
  - we often learn best from our mistakes
  - if you're not making mistakes, you're not challenging yourself

- Don't expect everything to be spelled out for you
  - real-world problems don't come that way
    - if there are detailed instructions for solving a problem, then there should already be a program that does it
  - world needs you for your intuition, creativity, & intelligence