



CS 463

Comparative Programming Languages

Introduction

Intro, Housekeeping

- Blackboard - mymason.gmu.edu
 - submit work, view grades
- Piazza – piazza.com
 - announcements, private correspondence, documents
- Gradescope – www.gradescope.com
 - Test administering, handwritten assignments
 - Built into our Blackboard course (Tools page)
- Syllabus / Schedule (on piazza)

Readings

- Our textbook is getting expensive – I've moved it to "recommended" readings, and you can use the library copies if you'd prefer. You can benefit from the materials in it, but don't purchase any "access" packs and other add-ons.
- Other readings are online for free (or paid for desk copies).

Keys to Success

- **Don't fall behind!** This course jumps around quite a bit, but we eventually revisit and combine topics.
- **Read** the reference material prior to lecture
- **Ask** questions
 - Silence denotes implicit understanding.
 - Please interrupt with comments/questions (chat) 😊
- **Utilize** available resources
 - piazza, office hours, etc.

Classroom Etiquette

- College is the start of your professional career—so act like it
- Don't distract/disrespect others (loud food, texting, surfing the web, chatting, reading the newspaper...)
- Let's discuss classroom rules and agree on them
 - Masking, safety, helping each other
 - Questions/interactions
 - Quiz timing

How to get an A

- Read all assigned reading before class.
- Attend all lectures.
- Get tools installed and experiment with them well before deadlines.
- Try assignments very early, in time to ask questions. Always turn in 100% working code.
- Go to office hours regularly with your questions! Also use the forums on piazza.
- Study early, often, and well for tests.

How to get a B

- Do pretty much all the reading.
- Only miss one or two lectures, and catch up with someone on what you missed.
- Start assignments well earlier than the night before, and get programs working.
- Occasionally use office hours, forum as needed.
- Study hard just before tests.

How to get a C

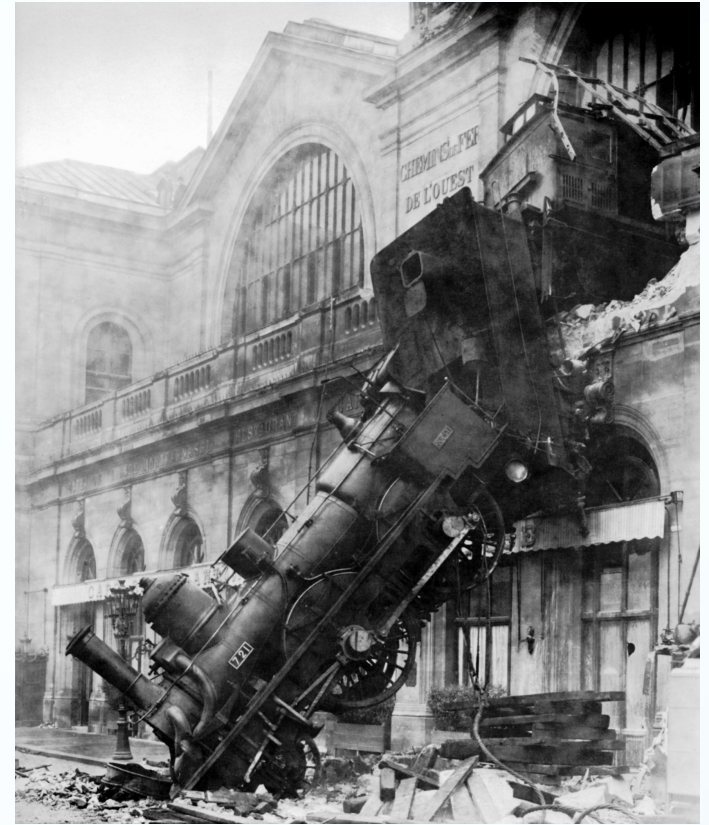
- Only miss one or two lectures. Goof off or get distracted occasionally in them, though.
- Do some of the reading, but not before class.
- Try hard on the assignments, but usually just the night before or in one 'power session'. No time to ask for help or clarification.
- "I don't have time for office hours".

How to get a D

- Miss a lot of lecture sessions.
- Skip most of the reading to save time, except when the work is too confusing.
- Start assignments at the last moment; spend much of your time fighting new tools.
- Realize about 2/3 into the semester that you're failing, and try really hard to catch up.
- Never come to office hours (except maybe when it's already too late!).

How to **Fail**

- Decide the first few weeks were easy, and stop coming to lecture.
- Skip all the reading because it's not graded.
- Start assignments the night before they're due, and not be able to ask questions.
- Cram for tests the night before, if at all.



Course Assumptions

- You have credit for CS 310, CS 330, and CS 367. You already know how to program in a procedural and object oriented style, and have perhaps seen the occasional niche language. (we're not re-teaching the basics—this course has its own content to cover)
- You are curious about language implementations, language features.
- You are an adult who understands that great grades are earned through hard, consistent work.
- You have interests beyond this course, classes besides this class, a life outside of Computer Science.

Course Assumptions

Many of you may have explored languages, libraries, and features that I have not; bring these experiences up, I want to hear about it too! 😊

Overview

If this were CS 363, we would just learn a new language every week and call it a semester.

- We will study language and type theory in more detail.
- We will spend more time on Haskell than other languages, as it has the opposite features of most languages you've learned.
- We have a (required) concurrency component
 - some of the course structure builds up to this
- Our assigned work will often be the same tasks in multiple languages.