CSCI 334:
Principles of Programming Languages

Lecture 1: Course intro

Instructor: Dan Barowy

Williams

Topics

What this course is **about**

What to **expect** in this course

Something to **think about** over the weekend

Every week:

1. Lab **due Sunday at 10pm**.
2. Reading check **due Wednesday at 10pm**.

Your to-dos

1. Lab 0, **due Sunday 9/11**.
   Grade scale:
   A if it is turned in.
   Otherwise… a different grade.
2. Check that you can clone your starter repository **today**.
   Only Lida/Kelsey can help you with account problems, and they are normal humans who stop working at 5pm on Friday.
3. Reading and reading check (see GLOW), **due Wednesday 9/14**.

If you don't remember your login…

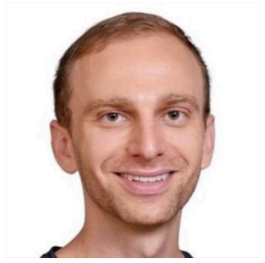
Email csaccounts@williams.edu

If you don't know/remember the TCL 312 door code…

Announcements

- First Colloquium this **Friday, Sept 9**





- Class field trip to WCMA on **Tuesday, Sept 13**
  Be sure to leave your bags, food, coats, etc. in the WCMA coat room. But please do bring pencil and paper.

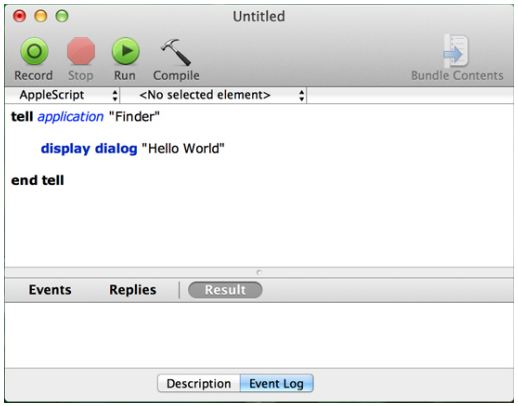
Why do I study programming languages?

A Bicycle for the Mind

A Bicycle for the Mind

A computer is a key part of this “bicycle”—but not all of it.

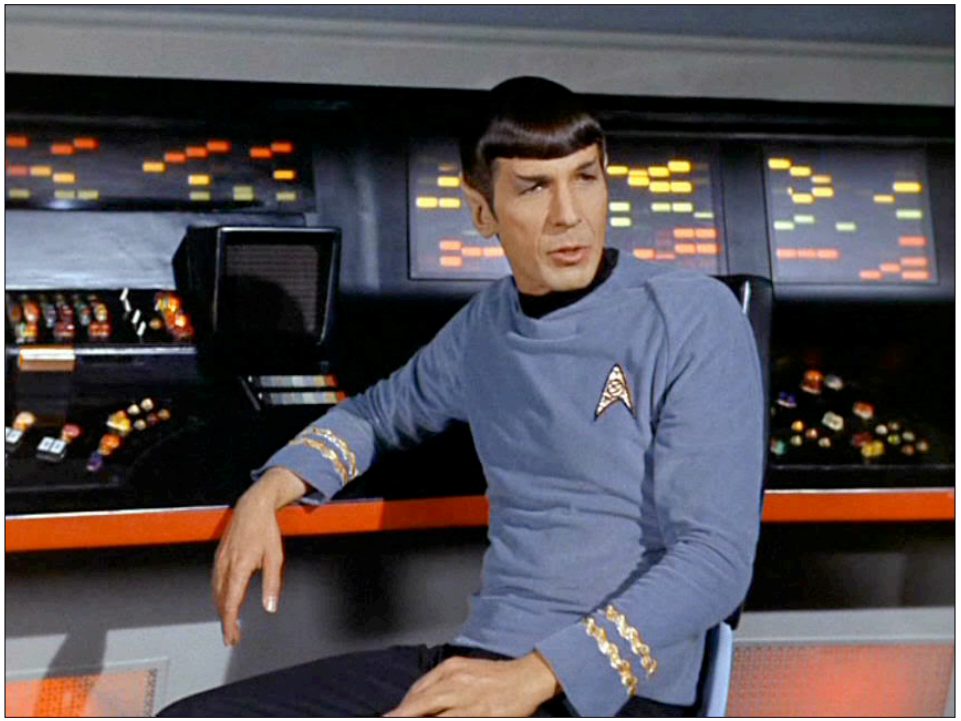
A Bicycle for the Mind



The other key part is a programming language.

AppleScript

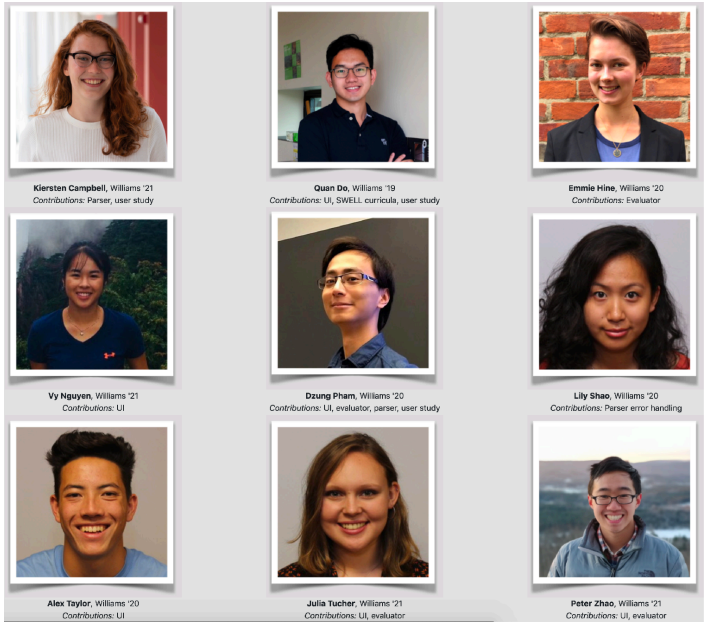
William R. Cook (1963-2021)



FlashFill

SWELL

Swell



Kiersten Campbell, Williams '21
Contributions: Parser, user study

Quan Do, Williams '19
Contributions: UI, SWELL curricula, user study

Emmie Hine, Williams '20
Contributions: Evaluator

Vy Nguyen, Williams '21
Contributions: UI

Dzung Pham, Williams '20
Contributions: UI, evaluator, parser, user study

Lily Shao, Williams '20
Contributions: Parser error handling

Alex Taylor, Williams '20
Contributions: UI

Julia Tucher, Williams '21
Contributions: UI, evaluator

Peter Zhao, Williams '21
Contributions: UI, evaluator

Riker: Always-Correct and Fast Incremental Builds from Simple Specifications

Authors:
Charlie Curtsinger, *Grinnell College*; Daniel W. Barowy, *Williams College*

Awarded Best Paper!

Abstract:
Build systems are responsible for building software correctly and quickly. Unfortunately, traditional build tools like make are correct and fast only when developers precisely enumerate dependencies for every incremental build step. Forward build systems improve correctness over traditional build tools by discovering dependencies automatically, but existing forward build tools have two fundamental flaws. First, they are incorrect; existing forward build tools miss dependencies because their models of system state are incomplete. Second, they rely on users to manually specify incremental build steps, increasing the programmer burden for fast builds.

This paper introduces Riker, a forward build system that guarantees fast, correct builds. Riker builds are easy to specify; in many cases a single command such as gcc *.c suffices. From these simple specifications, Riker automatically discovers fast incremental rebuild opportunities. Riker models the entire POSIX filesystem—not just files, but directories, pipes, and so on. This model guarantees that every dependency is checked on every build so every output is correct.

We use Riker to build 14 open source packages including LLVM and memcached. Riker incurs a median overhead of 8.8% on the initial full build. On average, Riker's incremental builds realize 94% of make's incremental speedup with no manual effort and no risk of errors.

Open Access Media

USENIX is committed to Open Access to the research presented at our events. Papers and proceedings are freely available to everyone once the event begins. Any video, audio, and/or slides that are posted after the event are also free and open to everyone. Support USENIX and our commitment to Open Access.

BibTeX

Curtsinger PDF

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

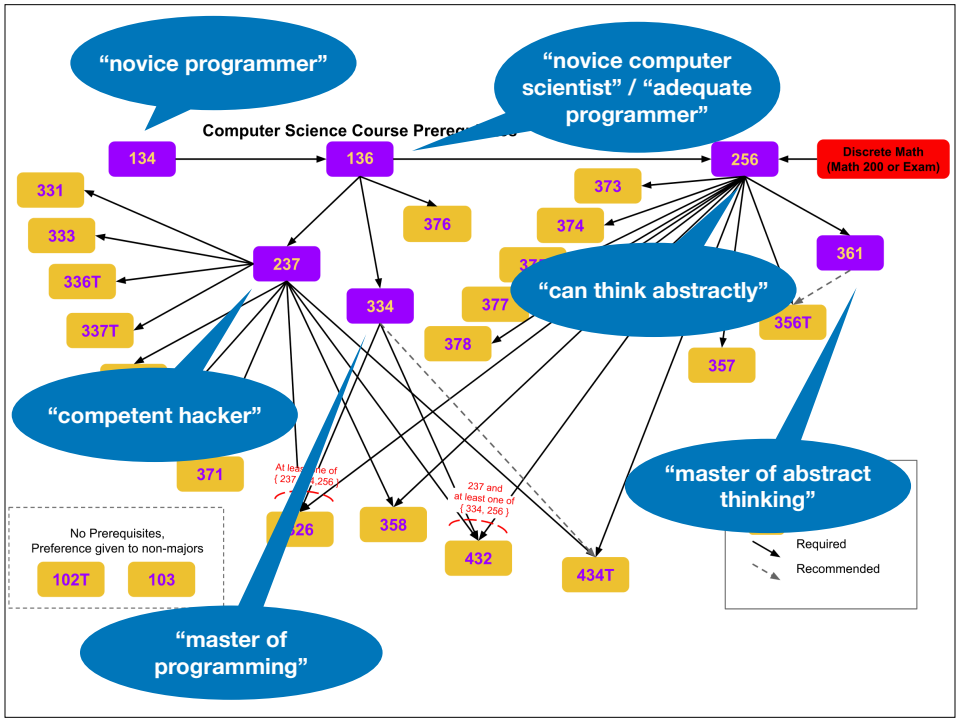
ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

Award: Best Paper

Class outcomes

1. Speak the “language of languages”
   a. understand the role of a language model
   b. evaluate fitness of language for purpose
   c. rapidly learn new languages
2. Add tools to your mental toolbox
   a. techniques for clear thinking
   b. become a (much) better programmer



Class outcomes

1. Speak the “language of languages”
   a. understand the role of a language model
   b. evaluate fitness of language for purpose
   c. rapidly learn new languages
2. Add tools to your mental toolbox
   a. techniques for clear thinking
   b. become a (much) better programmer
3. **Be your favorite class!**

Administrivia

Syllabus

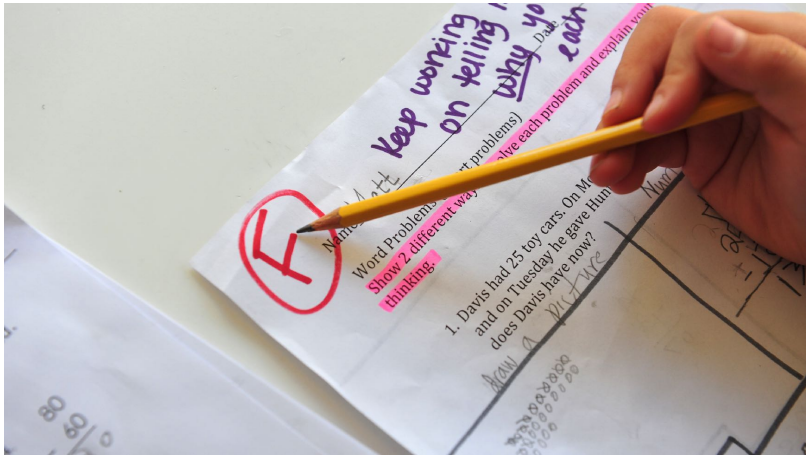
Weekly assignments

1. Weekly **reading checks**, due **Wednesday by 10pm** on GLOW.
   (~1 hour, on average, including the reading)
2. Weekly **labs**, due **Sunday by 10pm**
   (~5 hours, on average, but with wide variance)

Grades

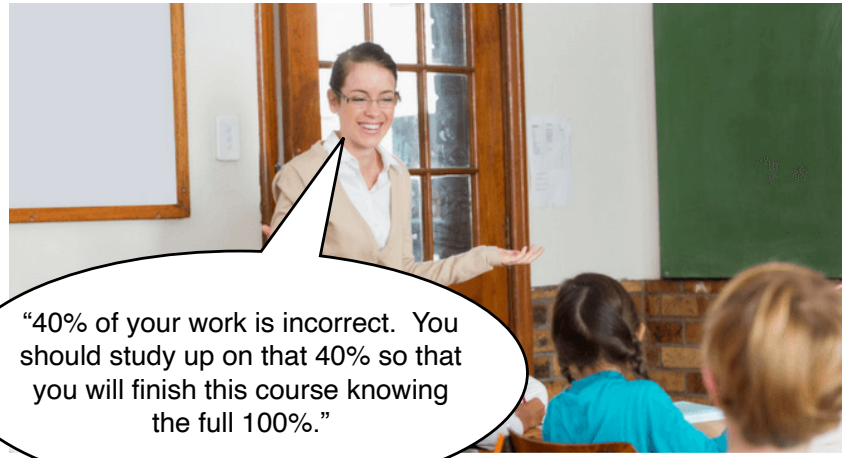
Midterm:	25%
Final Project:	25%
Homework assignments:	30%
Quizzes:	10%
Attendance:	10%

Let's discuss grades for a minute…



Let's say you get a 60% on your exam.

What your grade **means**.



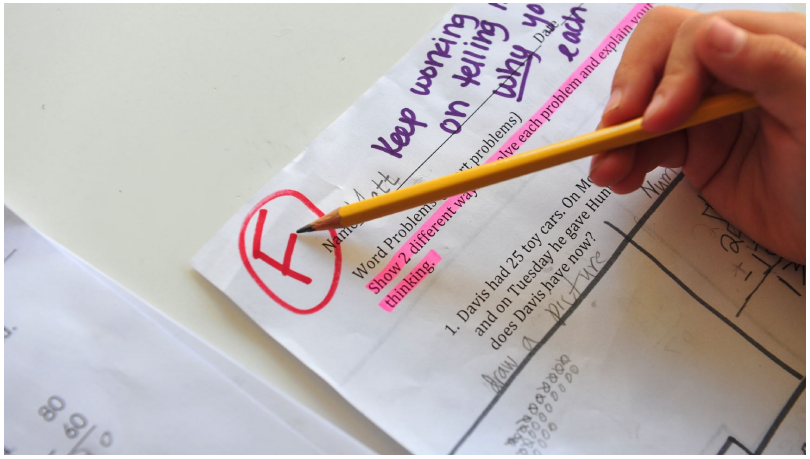
What your grade **does not mean**.

Do you feel…

Surprised? Embarrassed?

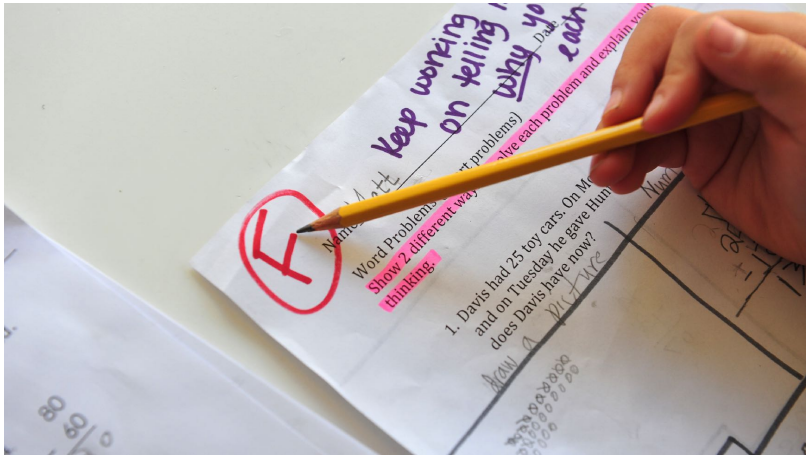
Your grade has zero bearing on whether I like you or not.

The purpose of a class



To turn weaknesses into a strengths.

The purpose of a grade



A way to identify a weakness.

You are capable of choosing how you respond to grades

"It is our choices that show what we truly are, far more than our abilities." —Dumbledore

If you

1. submit your work **on time**,
2. pay attention to **feedback**, and
3. diligently **correct your mistakes**,

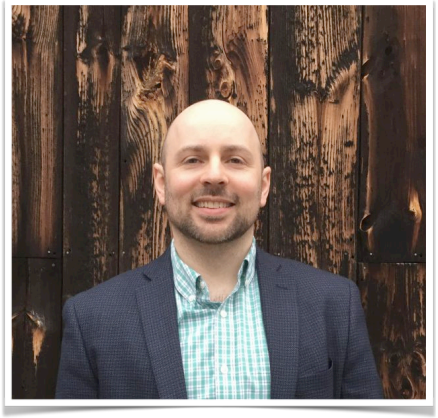
you have **little to worry about** in this class.

You have a lot of control in this formula!

Midterm:	25%
Final Project:	25%
Homework assignments:	30%
Quizzes:	10%
Attendance:	10%

Help

Me



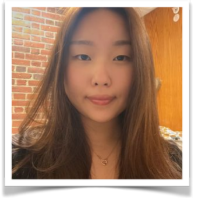
Our Wonderful TAs

Ruby

Rijul

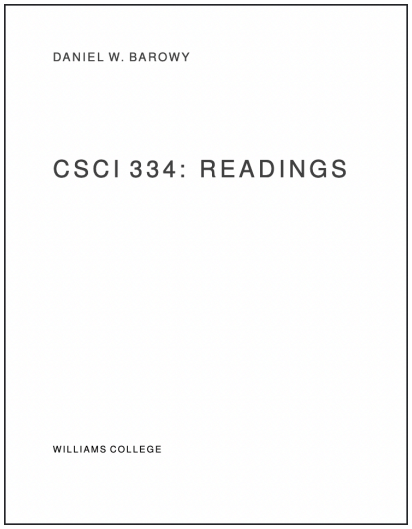
Rito

Paul

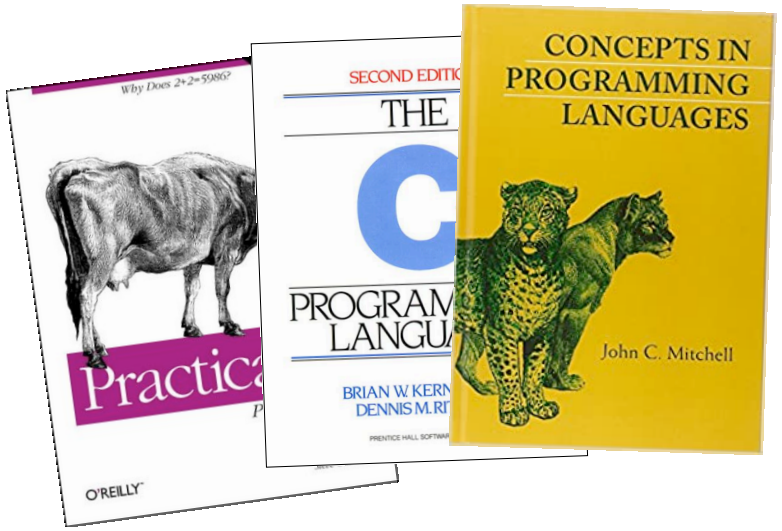


Evelyn

Course packet



Books on reserve in Schow



Course readings

Possibly the most important part of the class.

Course readings

Reading checks

Anonymous grading

Homework late policy:

Late Work. You are expected to turn in all assignments in a timely manner to receive full credit. Please contact me ahead of time to discuss the matter if you foresee an issue that prevents timely submission. Without prior arrangement, late assignments will be penalized at a rate of **20% per day**.

Something better: homework resubmissions

Resubmissions. The assignments in this course are designed to challenge you. You may find that occasionally, you do not do as well on an assignment as you had hoped. To encourage you to revisit and master this material, I allow up to two assignment resubmissions during the semester. This policy includes labs 1–9 and the midterm exam, but not quizzes, the final lab, or the final project.

A resubmission will be accepted at the discretion of the course instructor and allows you to earn back **up to 50% of the missing points**. For example, if you received a 75% on an assignment, you may earn up to 87.5% upon resubmission.

Resubmissions must be submitted in the following manner:

1. They must be submitted before the end of the final exam reading period.
2. They must include both the original work and the new submission.
3. They must be accompanied with a typed document, written in plain language, that explains, for every misunderstanding:
   (a) what the error is in the original work,
   (b) how you fixed the error, and
   (c) why the new version is correct.

Please note that resubmissions must be typed or they will not be accepted. Detailed instructions for submitting a resubmission will be distributed via a separate handout.

Honor Code

COVID

Gitlab

https://evolene.cs.williams.edu/

Course website

https://williams-cs.github.io/cs334-f22-www/

Course Organization

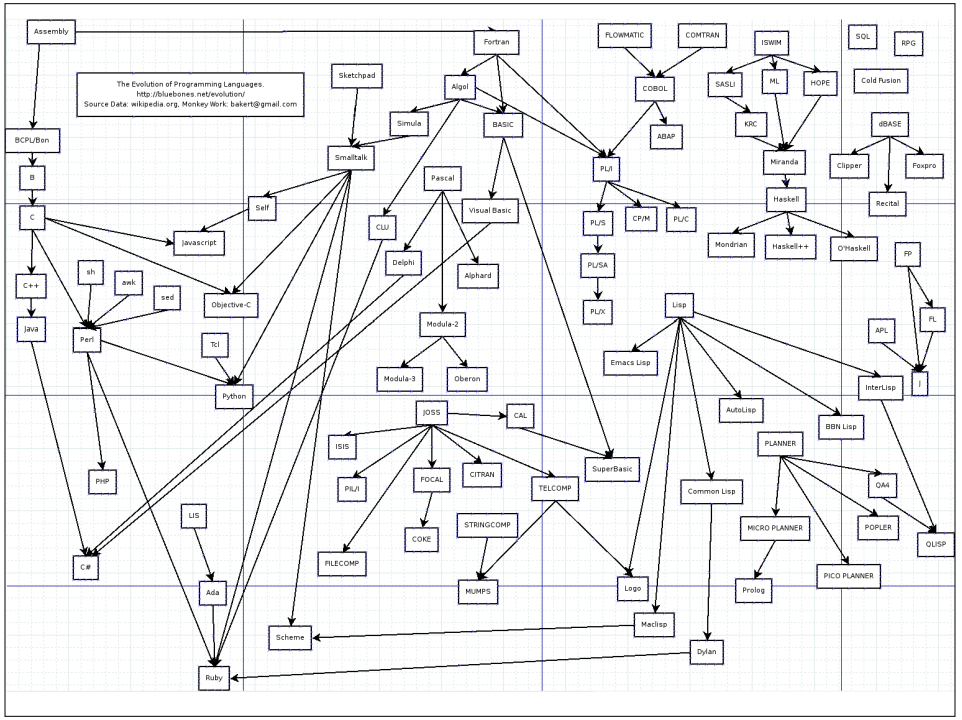
First half

- language models
- theoretical foundations
- functional vs. imperative languages
- new ways of thinking

Course Organization

Second half

- language architecture
- object oriented programming
- projects!



You are going to get your hands dirty

I always want you to succeed

Our notions for success may not always be the same, but I promise you: I do not assign busy work.

I always care about what you think

1. Optional feedback on assignments
2. Optional, anonymous feedback form on course website

Is this a programming language?

https://openai.com/dall-e-2/

For the weekend:

What is a programming language?

Recap & Next Class

Today:

Course goals

Course structure

Next class:

F#