Welcome and Syllabus!

Sam McCauley and Dan Barowy

February 4, 2022

Course Materials and Tools

- Course website
  - `http://cs.williams.edu/~cs136/index.html`
  - Syllabus, schedules, office hours, *readings*, links to (virtually) all content
- Gitlab
  - `evolene.cs.williams.edu`
  - Used for lab submissions, starter code, grading, etc.

Office Hours!

- Professor office hours are now on the website.
  - Sam Mon 1–3, Tue 2–4, Wed 1–2; Dan Mon 3–4:30
  - On this zoom link until Wednesday; in person after that

Office Hours!

- Professor office hours are now on the website.
  - Sam Mon 1–3, Tue 2–4, Wed 1–2; Dan Mon 3–4:30
  - On this zoom link until Wednesday; in person after that

- TA office hours are only in person; start after Lab 1

Office Hours!

- Professor office hours are now on the website.
  - Sam Mon 1–3, Tue 2–4, Wed 1–2; Dan Mon 3–4:30
  - On this zoom link until Wednesday; in person after that
- TA office hours are only in person; start after Lab 1
- Will be posted soon. Lots of availability on Monday and Tuesday evenings

Tools for Code

- Atom
  - You’re already a little familiar
  - Primary way you’ll write code this semester

Tools for Code

- Atom
  - You’re already a little familiar
  - Primary way you’ll write code this semester
- emacs
  - Very useful tool in the programmer’s toolbox
  - Lightweight, works over terminal
  - We’ll talk about how it’s used sometimes in class
  - You are not required to code with it however.

Tools for Code

- Atom
  - You're already a little familiar
  - Primary way you'll write code this semester
- emacs
  - Very useful tool in the programmer's toolbox
  - Lightweight, works over terminal
  - We'll talk about how it's used sometimes in class
  - You are not required to code with it however.
- terminal
  - You'll be using to compile and run Java programs
  - We'll talk a bit about how to make it easy to use

Why Take CS136?

- To learn about:
  - Data Structures
    - Effective ways to store and manipulate data
  - Advanced Programming
    - Combine data structures, programming techniques, and algorithmic design to write programs that solve interesting and important problems
  - Basics of Algorithm Analysis
    - Measuring algorithm complexity
    - Establishing algorithm correctness

Goals

- Identify basic data structures
  - list, stack, array, tree, graph, hash table, and more
- Implement these structures in Java
- Learn how to evaluate and visualize data structures
  - Linked lists and arrays both represent lists of items
  - Different representations of data
  - Different algorithms for manipulating/accessing/storing data
- Learn how to design larger programs that are easier to modify, extend, and debug
- **Have fun!**

Course Outline

- Java review
- Basic structures
  - Lists, vectors, queues, stacks
- Advanced structures
  - Graphs, heaps, trees, dictionaries
- Foundations (throughout semester)
  - Vocabulary
  - Analysis tools
  - Recursion & Induction
  - Methodology

Why Java?

- Java is (still) popular!

Why Java?

- Java is (still) popular!
- Object-oriented—good for large systems

Why Java?

- Java is (still) popular!
- Object-oriented—good for large systems
- Good support for abstraction, extension, modularization

Why Java?

- Java is (still) popular!
- Object-oriented—good for large systems
- Good support for abstraction, extension, modularization
- Automatically handles low-level memory management

Why Java?

- Java is (still) popular!
- Object-oriented—good for large systems
- Good support for abstraction, extension, modularization
- Automatically handles low-level memory management
- Very portable

Common Themes in This Course

- Identify data for problem
- Identify questions to answer about data
- Design data structures and algorithms to answer questions *correctly* and *efficiently*
  - Note: not all correct solutions are efficient
  - And vice versa!
- Implement solutions that are robust, adaptable, and reusable
- Example: Shortest Paths in Networks

Example: Shortest paths

Finding Shortest Paths

- The data: road segments
  - Road segment: Source, destination, length (weight)
- The question
  - Given source and destination, compute the shortest path from source
- The algorithm: Dijkstra's Algorithm
- The data structures (spoiler alert!)
  - Graph: holds the road network in some useful form
  - Priority Queue: holds not-yet-inspected edges
  - Also uses: Lists, arrays, stacks, ...
- A demo....

Let’s take a look at the syllabus

CS Mentoring!

- New program in CS

CS Mentoring!

- New program in CS
- Collaboration between UnICS and CoSSAC

CS Mentoring!

- New program in CS
- Collaboration between UnICS and CoSSAC
- Monthly meetings between alumni mentor and student mentee

CS Mentoring!

- New program in CS
- Collaboration between UnICS and CoSSAC
- Monthly meetings between alumni mentor and student mentee
- To participate, submit this form: `https://forms.gle/H2p4hUsJ45KVjXzG9`

CS Mentoring!

- New program in CS
- Collaboration between UnICS and CoSSAC
- Monthly meetings between alumni mentor and student mentee
- To participate, submit this form: `https://forms.gle/H2p4hUsJ45KVjXzG9`
- Contact UnICS or CoSSAC with questions

Let’s take a look at Java programming

- Next couple days: (re)introduction to Java

Let’s take a look at Java programming

- Next couple days: (re)introduction to Java
- Some of you have Java experience; some don’t

Let’s take a look at Java programming

- Next couple days: (re)introduction to Java
- Some of you have Java experience; some don’t
- Goal: quickly get everyone up to speed for the first couple labs

Let’s take a look at Java programming

- Next couple days: (re)introduction to Java
- Some of you have Java experience; some don’t
- Goal: quickly get everyone up to speed for the first couple labs
- If you are less comfortable with Java, be sure to keep up with readings and ask questions early on!