Lecture 1

Introduction

- People
- Course Overview
- Java
- Next Steps
People
Students
All students entering the course have the following background:

- Completed CSCI 134 (in Python), or
- Passed the Computer Science A AP exam (in Java), or
- Programming Experience (i.e., internship).
Faculty

Duane Bailey

● Wrote the textbook.
● Wrote the structures software library.
● Teaching 2 Sections + 3 Labs.
● Look for mathematical artwork in the Schow Library.
● [cs.williams.edu/~bailey/](cs.williams.edu/~bailey/)  bailey@cs.williams.edu

Aaron Williams

● First time teaching this course.
● Some students will know more about Java!
● Teaching 1 Section + 2 Labs.
● Research includes combinatorial algorithms, computational complexity, video game history.
● [cs.williams.edu/~aaron](cs.williams.edu/~aaron)  aw14@williams.edu
Staff

Lida Doret

- Instructional Support.
- *git* magic!

Note: Our team supports the Computing Environment (including lab computers and servers) and *not* your personal computer.

Lida Doret
Course Overview
# CSCI 136

## Data Structures and Advanced Programming

<table>
<thead>
<tr>
<th>Lectures</th>
<th>Labs</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Home</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Instructors:
- Duane A. Bailey (email: bailey), TPL 306
- Aaron Williams (email: aw14), TBL 309A

### Technical Support:
- Lida Doret (email: lpd2), TCL 205

### Web Site:
http://www.cs.williams.edu/~cs136 (this page!)

### Course Calendar:
https://tinyurl.com/cs136-calendar

### Lecture:
- Schow 030B, MWF 9-9:50am (§ 1), 10-10:50am (§ 2), or 11-11:50am (§ 3)

### Labs:
- All labs are in Chemistry 217a
- W 1:10-2:25pm, with Bailey (§ 7); or
- W 2:35-3:50pm, with Williams (§ 8); or
- R 9:55-11:10am (§ 4) or 1:10-2:25pm (§ 5), with Bailey; or
- R2:35-3:50pm (§ 6), with Williams.

### Textbook:
- Duane Bailey's *Java Structures (Root-7 Edition)*, here (book resource page is here)

### TAs:
- Milo Chang, Kary Chen, Samuel Chistolini, Diego Esparza, Gaurnett Flowers,
- Nolan Holley, Emma Neil, Saul Richardson, and Ye Shiu

### TA Hours:
- Posted to the Course Calendar

Information is found on the course webpage: [www.cs.williams.edu/~cs136](http://www.cs.williams.edu/~cs136)
Computer Science 136: Data Structures & Advanced Programming

Prof. Duane Bailey
bailey@williams.edu

Prof. Aaron Williams
aw14@williams.edu

Fall 2021

Technical Support: Lida Doret lpd2@williams.edu
Course web site: www.cs.williams.edu/~cs136
Textbook: www.cs.williams.edu/~bailey/JavaStructures
Lectures: Students are enrolled in one of the following lecture sections, which meet in Schow 030B.
  • MWF 9:00-9:50am, 10:00-10:50am, or 11:00-11:50am
Lab Sections: Students are enrolled in one of the following lab sections, which meet in TCL 217A.
  • Wednesday 1:10-2:25pm or 2:35-3:50pm
  • Thursday 9:55-11:10am, 1:10-2:25pm, or 2:35-3:50pm
Instructor Hours: See course calendar
Assistants: Milo Chang, Kary Chen, Samuel Chistolini, Diego Esparza, Gaurmet Flowers,
  Nolan Holley, Emma Neil, Saul Richardson, Ye Shu
TA Hours: See course calendar

Description. This course couples work on program design, analysis, and verification with an introduction to the study of data structures. Data structures capture common ways in which to store and manipulate data, and they are important in the construction of sophisticated computer programs. We will use the Java programming language in class and for the assignments.

Students will be expected to write several programs, ranging from very short programs to more elaborate ones. Since one of our goals in this course is to teach you how to write large, reliable programs composed from reusable pieces, we will be emphasizing the development of clear, modular programs that are easy to read, debug, verify, analyze, and modify.

You will be carrying out your programming assignments on laboratory computers in the department. All of the software tools you will need for this course are installed on these machines. Our first lab will be devoted to guiding you through the workflow we expect you to use throughout the rest of the semester.

Course syllabus: www.cs.williams.edu/~cs136/syllabus.pdf
<table>
<thead>
<tr>
<th>Monday</th>
<th>Wednesday</th>
<th>Lab</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sept. 6</td>
<td>—</td>
<td>—</td>
<td>1. Introduction</td>
</tr>
<tr>
<td>Sept. 13</td>
<td>2. Java Basics (Ch. 0,1,B)</td>
<td>3. Organization</td>
<td>4. Associations (Ch. 1,2)</td>
</tr>
<tr>
<td>Sept. 20</td>
<td>5. Vectors (Ch. 3,4)</td>
<td>6. Complexity (Ch. 5)</td>
<td>Vectors</td>
</tr>
<tr>
<td>Sept. 27</td>
<td>8. Recursion II (Ch. 5)</td>
<td>9. Lists I (Ch. 7,9)</td>
<td>Recursion</td>
</tr>
<tr>
<td>Oct. 4</td>
<td>11. Sorting &amp; Search (Ch. 6)</td>
<td>12. Sorting II. (Ch. 6)</td>
<td>Linked Lists</td>
</tr>
<tr>
<td>Oct. 11</td>
<td>Reading Period</td>
<td>Midterm Prep</td>
<td>13. Stacks (Ch. 10)</td>
</tr>
<tr>
<td>Oct. 18</td>
<td>14. Queues (Ch. 10)</td>
<td>15. Ord. Structs. (Ch. 11)</td>
<td>Stacks</td>
</tr>
<tr>
<td>Oct. 25</td>
<td>17. Lambdas (TBD)</td>
<td>18. Generation (TBD)</td>
<td>Skip Lists</td>
</tr>
<tr>
<td>Nov. 1</td>
<td>19. Trees I (Ch. 12)</td>
<td>20. Trees II (Ch. 12)</td>
<td>Iteration</td>
</tr>
<tr>
<td>Nov. 8</td>
<td>22. Heaps (Ch. 13)</td>
<td>23. Search Trees (Ch. 14)</td>
<td>Trees</td>
</tr>
<tr>
<td>Nov. 15</td>
<td>25. Maps &amp; Dicts. (Ch. 15)</td>
<td>26. Hashtables (Ch. 15)</td>
<td>Hashing</td>
</tr>
<tr>
<td>Nov. 22</td>
<td>29. TBD</td>
<td>31. Graphs (Ch. 16)</td>
<td>Graphs</td>
</tr>
<tr>
<td>Nov. 29</td>
<td>30. Graphs (Ch. 16)</td>
<td>34. TBD</td>
<td>32. Graphs (Ch. 16)</td>
</tr>
<tr>
<td>Dec. 6</td>
<td>33. TBD</td>
<td>—</td>
<td>35. Review</td>
</tr>
</tbody>
</table>

**Quiz dates:** Evening of October 14 and during final exam period.

*(Tentative) Course Schedule from the Syllabus.*
Grading.
Computing Environment.
Textbook.

- We will use an electronic version.
- Older (but not that old!) versions have been printed in hardcover.
Succeeding in the Course

Below are some steps that will ensure that you are successful in the course.

- **Read** the textbook.
  - The readings are given in the course schedule (see previous slide).
  - Try to understanding everything — don’t skim.

- **Be engaged** during lecture.
  - Refocus yourself when necessary.
  - Sit closer to the front if that helps.

- **Prepare** for the labs.
  - Start reading and working on the lab before your lab section. Handouts are available on Tuesdays.
  - Work through difficult parts of the lab during lab time.
  - Continue working on the lab when the lab time is finished.

- **Use the course and college resources.**
  - TA hours.
  - Instructor office hours.
  - Conversations with other students.
  - Check syllabus for additional support.

- **Have fun!**
Java
Saying hello in Python (left) and Java (right).
Java often feels like a bureaucratic language.

- There are advantages and disadvantages to this style.
I am **personally** excited to learn more about Java for many reasons:

- **Retrogame archeology.** Java is the language used in many pre-iPhone mobile devices.
- **History.** As an undergraduate student in the late-1990s, I had three internships at Corel.
Hello, World!
Our first program
Live Coding: Hello, World! (Hello.java)

- Write a Java program that prints something to the console.
- Compile and run the program from the command-line.
- Discuss the syntax and various keywords.
  - class, public, static, void, main, System.out, {}, ;
Live coding from class.

```java
class Hello {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

```bash
$ ssh aaron@lohani.cs.williams.edu
aaron@lohani.cs.williams.edu's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:     https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Fri 10 Sep 2021 12:13:43 PM EDT

System load: 0.17
Usage of /: 60.0% of 439.11GB
Memory usage: 2%
Swap usage: 0%
Temperature: 38.0°C
Processes: 820
Users logged in: 6
IPv4 address for ens1f0: 137.165.120.103

* Super-optimized for small spaces – read how we shrank the memory footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

48 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradeable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri Sep 10 11:25:23 2021 from 137.165.120.103
$ cd cs136/live/
$ nano Hello.java

$ javac Hello.java
$ java Hello
Hello, World!
Challenge: Recall the Program

Try to recreate Hello, World!

- Write the program down on paper, or on your computer.
- Do not refer to any of your notes.
- (Time Permitting) Compare your program with a neighbor.
Knock, Knock: Command-Line Arguments
Live Coding: Knock, Knock (**Knock.java**)

- Write a Java program that reads and prints command-line arguments.
- Use an if statement to put a comma between the arguments.
- Use two different types of for loops to iterate over the arguments.
- *(Time Permitting)* Discuss the *nano* editor.
- Discuss additional syntax and keywords.
  - `String, int, [], .length, :`
```java
class Knock {
    public static void main(String[] args) {
        System.out.println("Knock, Knock");
        System.out.println("Who's There?");

        // Print the arguments with commas between them.
        for (int i = 0; i < args.length; i++) {
            System.out.print(args[i]);
            if (i < args.length - 1) {
                System.out.print(",");
            }
        }
        System.out.println("\n");

        // Print the arguments again using a different style.
        for (String arg : args) {
            System.out.print(arg);
        }
        System.out.println("\n");
    }
}
```

Live coding from class (with some extra comments and printing added).
nano:
the simple terminal editor
nano is one of the the simplest terminal-based text editors.

- Learn more: man nano or tldr nano (online version: tldr.sh) or Ctrl+g in the program.
- For configuration refer to the ~/.nano folder and the ~/.nanorc file.
- Our Unix machines have Version 4+ but the Mac machines may only have Version 2.
- Other terminal options: emacs or vi(m). Atom is an excellent non-terminal text editor.
Next Steps
Next Steps
In this lecture we gave an overview of the course, and wrote our first Java programs.

Lecture 2: Java Basics
● Classes, objects, ...

Lecture 3: Organizing Code
● Tools: git, ssh, ...

Lab 0: Computing Environment
● Getting used to how our labs will work ...

Lecture 4: Associations
● Our first real data structure ...
● The structure package ...

Note: Duane will be substituting for me during Lecture 2.