Records & Data Analysis

Exam Reminders

- Plan to take your exam during the section for which you are registered.
- Take the practice exam soon!
- All students who require SDS accommodation to take the exam should schedule their exam through the Weingarten Testing Center ASAP.
    - Any time on Monday is acceptable as a testing time.
- The exam only covers material up until functions & searching. Material covered Wednesday and today will be covered on HW04 and Exam 2.

Last Time

Discussion about **Data Oriented Programming** as a way of thinking about how we write our programs

- Key takeaway: try to write programs that separate code & data
- Less important stuff for CIS 1100: ideas about efficient organization of computer memory, discussions about how programs relate to the hardware

Last Time

Introduction of the **Record Type** in Java as a way to...

- Define a new data type that can be used in our program
- Specify *succinctly* what the data looks like that the program intends to manipulate or analyze

```
public record OceanSample(double lat, double lon, double temp, String time,
                          double o2Pct, double ironMass, double biomass) {}
```

~one line to describe what samples of ocean climate data look like, for example!

Data Analysis & Data Science

Data Science is a multi-disciplinary domain that overlaps significantly with Computer Science

- *In Common:* data storage; algorithms for aggregation, sorting, combining data; database management
- *Outside of CS:* statistics for analysis, domain-specific knowledge about data application, generating the data itself

In fact, most of you here outside of CIS majors will be most interested in these skills

This Class & Data Analysis

This is a *computer science* course designed to teach *programming* in Java.

- We will spend some time now & on HW04 practicing data analysis techniques
  - These exercises are some of the most applicable for folks in non-CIS fields
- We will move past our data focus into *object oriented programming* for a while before returning at the end to a more "data-y" setting

A Model Pipeline

1. Obtaining the Data
    - Here, I do this for you.
2. Understanding the Data
    - Nobody else can ever do this for you!
3. Parsing the Data
4. Cleaning the Data
    - In the interest of time, the data that I give you to work with is pretty "clean"
    - No missing entries, no weird formats
5. Analyzing the Data

Worked Example: Books!

I'm vain, so we're going to use my personal data: my collection of books read from Goodreads. (follow me?)

We'll use this data to build a recommender system

- "I heard about this author, can you recommend me her best book?"
- "What's the best book from last year?"

Understanding the Data

What's going on here?

What do we have to work with?

```
Voices in the Evening
Natalia Ginzburg
1952 170 3.76
The Dry Heart
Natalia Ginzburg
1947 88 3.99
Childhood / Youth / Dependency (The Copenhagen Trilogy, #1-3)
Tove Ditlevsen
1967 371 4.36
In the Eye of the Wild
Nastassja Martin
2019 128 3.96
Kudos
Rachel Cusk
2018 236 3.91
Jack (Gilead, #4)
Marilynne Robinson
2020 309 3.86
```

Understanding the Data

For each data point (Book), we have:

- title
- author
- year, page count, rating

```
Voices in the Evening
Natalia Ginzburg
1952 170 3.76
The Dry Heart
Natalia Ginzburg
1947 88 3.99
Childhood / Youth / Dependency (The Copenhagen Trilogy, #1-3)
Tove Ditlevsen
1967 371 4.36
In the Eye of the Wild
Nastassja Martin
2019 128 3.96
Kudos
Rachel Cusk
2018 236 3.91
Jack (Gilead, #4)
Marilynne Robinson
2020 309 3.86
```

Understanding the Data

For each data point (Book), we have:

- title
  - String
- author
  - String
- year, page count, rating
  - int, int, double

Understanding the Data

For each data point (Book), we have:

- title
  - String
- author
  - String
- year, page count, rating
  - int, int, double

```
public record Book(String title, String author,
                   int year, int pages, double rating)
```

Parsing the Data

We need to write a function (in this case `main`) that can take data in a file and read it into `Book` records in our program.

```
String filename = args[0];
In reader = new In(filename);

int numBooks = reader.readInt();
System.out.println(numBooks);

Book[] books = new Book[numBooks];
for (int i = 0; i < numBooks; i++) {
    reader.readLine(); // proceed to next line...
    String title = reader.readLine().trim();
    String author = reader.readLine().trim();
    System.out.println(title);
    int year = reader.readInt();
    int pages = reader.readInt();
    double rating = reader.readDouble();

    books[i] = new Book(title, author, year, pages, rating);
}
```

Analyzing the Data

The types of analysis we'll do correspond to the kinds of questions we want to answer.

- "*I heard about this author, can you recommend me her best book?*"
  - "best" → highest `rating`
  - "her best" → only consider books with proper `author` value
  - This is a "find a maximum value in array" problem!
- "*What's the best book from last year?*"
  - "best" → highest `rating`
  - "from last year" → only consider books with proper `year` value
  - This is the same exact problem!!

Analyzing the Data

Observe: lots of questions you want to answer are just different versions of the same thing

- find the max...
- find the min...
- find the sum...
- find the average...
- find the first...
- find the last...

Analyzing the Data

Another common question: *find all data points that match a criteria*, e.g.:

- "What have you read by this author?"
- "What kinds of books do you usually read in the winter?"

Similar to finding a max/min/sum/etc., but we have to collect *multiple* results in an array to answer the question.