CSE 142 vs CSE 143

CSE 142
• You learned how to write programs and decompose large problems with:
  • Print statements
  • Methods
  • Control Structures
    • loops, if/else
  • File I/O
  • Arrays
  • Objects

CSE 143
• You learned to solve more complex tasks efficiently
  • Data structures to organize and model data
  • Algorithms for solving common tasks
  • More advanced language features
  • Abstractions are important!
Road Map

**CS Concepts**
- Client/Implementer
- Efficiency
- Recursion
- Regular Expressions
- Grammars
- Searching / Sorting
- Backtracking
- Hashing
- Huffman Compression

**Java Language**
- Exceptions
- Interfaces
- References
- Comparable
- Generics
- Inheritance / Polymorphism
- Abstract Classes

**Data Structures**
- Lists
- Stacks
- Queues
- Sets
- Maps
- Priority Queues

**Java Collections**
- Arrays
- ArrayList
- LinkedList
- Stack
- TreeSet / TreeMap
- HashSet / HashMap
- PriorityQueue
Major themes

- Abstraction
  - Leverage existing components without understanding details
  - Create components that can be used as black boxes

- Design tradeoffs
  - Algorithm analysis - scalability and growth
  - Keeping code easy to read for maintainability

- Recursion
  - Reason about problems in terms of self-similarity
  - Write very short code to achieve complex behaviors

- Art – "A programmer who subconsciously views himself as an artist will enjoy what he does and will do it better." (Knuth)
What project?

- Little text-processing applications
  - identify lines above 100
  - remove line-breaks

- Add a GUI to the random sentence generator

- Automate chemistry, physics, calculus problems, etc

- Find quotes by keyword in books

- What are you currently doing that a computer could do?

- List of some project ideas
What language?

• Expanding your Java knowledge with a project is valuable

• Pick a project, see what language is most appropriate
  • iOS: Swift
  • Android: Java
  • Client-side web: Javascript (many frameworks to choose from)
  • Beautiful visuals: Processing
  • Data Processing: Python
  • Data Management: SQL
  • Embedded systems: C / C++

• Learn a new paradigm
  • Functional languages: Racket, Haskell, Scala, (now, Java 8!)
Leveraging existing code

- Processing language

- Building games
  - [http://lwjgl.org/](http://lwjgl.org/)
  - [http://jbox2d.org/](http://jbox2d.org/) (with physics!)

- Processing biological data
  - [http://biojava.org/wiki/Main_Page](http://biojava.org/wiki/Main_Page)

- Accessing Facebook data
  - [http://restfb.com/](http://restfb.com/)

- Making music
  - [http://www.jfugue.org/](http://www.jfugue.org/)
Courses?

- CSE non-majors
  - CSE 154: Web Programming
  - CSE 373: Data Structures and Algorithms
  - CSE 374: Programming Concepts and Tools (C/C++, Linux, …)
  - CSE 131: Digital Photography
  - CSE 160: Intro to Python
  - CSE 460: Animation Capstone (open to all majors)

- CSE majors
  - CSE 311: (Mathematical) Foundations of Computing
  - CSE 332: Data Abstractions (Data Structures and Algorithms)
  - CSE 331: Software Design and Implementation
  - CSE 341: Programming Languages
  - CSE 344: Intro to Data Management (and databases)
  - CSE 351: Hardware/Software Interface

- INFO, AMATH, HCDE, DXARTS, …
Beyond programming

- Mind-controlled robots
  - [http://www.youtube.com/watch?v=TQ7EOpPNQyw](http://www.youtube.com/watch?v=TQ7EOpPNQyw)

- Muscle-controlled interfaces
  - [http://www.youtube.com/watch?v=pktVSTwC8qo](http://www.youtube.com/watch?v=pktVSTwC8qo)

- 3D models from pictures
  - [http://www.youtube.com/watch?v=25Yifq70eIY](http://www.youtube.com/watch?v=25Yifq70eIY)

- Face aging
  - [http://www.youtube.com/watch?v=fLQtssJDMMc](http://www.youtube.com/watch?v=fLQtssJDMMc)

- Animation
  - [http://www.youtube.com/watch?v=b4kkPILdMvI](http://www.youtube.com/watch?v=b4kkPILdMvI)

- Security
Weekly meetings

- Change – technologies for low-income regions
  - [http://change.washington.edu/](http://change.washington.edu/)

- Dub – human-computer interaction and design
  - [http://dub.washington.edu/](http://dub.washington.edu/)
Computer Science Books

The Hidden Language of Computer Hardware and Software

Charles Petzold

9 Algorithms That Changed the Future

The Ingenious Ideas That Drive Today's Computers

John MacCormick

Dear Data

Giorgia Lupi

Foreword by Maria Popova

Leena Trivedi
Computing & Jobs


- Computer occupations (15-1100)
- Engineers (17-2000)
- Life scientists (19-1000)
- Physical scientists (19-2000)
- Social scientists and related workers (19-3000)
- Mathematical science occupations (15-2000)

Data from the spreadsheet at http://www.bls.gov/emp/ind-occ-matrix/occupation.xlsx
Internships

- Various career fairs around campus.

- Start looking early!

- Cast a broad net and interview lots of places

- For those just starting out
  - Microsoft Explorer Program –
  - Google Engineering Practicum -
Roles in Industry

- **Software Developer/Software Engineer**
  - Builds and designs software
  - Includes designing and engineering architecture of a software system as well as programming

- **Product Manager (PM)**
  - Designs and makes decisions regarding the overall product
  - Works with people across disciplines at the company
  - Role can be different at different companies

- **Test/QA**
  - Write and design tests of the product

- **Site Reliability Engineer (SRE)**
  - Responsible for ensuring that systems and services are available and responsive
Small vs Big Company?

- **Small Company**
  - Lots of autonomy and impact within the company
  - Often move quickly
  - Breadth – get to work on many projects and with many types of people

- **Large company**
  - Large data sets, impact many users
  - Lots of support and infrastructure to do your job well
  - Depth – get to focus on specific areas of a project
What Do I Do?

- I’m a graduate student at the Paul G Allen Center for Computer Science. Topics in CS that interest me:
  - Machine Learning
  - Data Visualization
  - Theoretical Computer Science
- Where I have interned
  - **Redfin**
    - Job: Full-stack engineer (worked on frontend and backend)
    - Frontend: Javascript (Dojo/React)
    - Backend: Java!
  - **Socrata**
    - Job: Mostly data science, a little of backend work on search
    - **Seattle City Data**
    - Machine Learning: Python
    - Search Backend: Scala + ElasticSearch
AMA
(ask me anything)