A Programming Approach to the CS Principles Data Task
(from The Beauty and Joy of Computing team)

UC Berkeley
Sr Lecturer SOE
Dan Garcia

UC Berkeley
Research Scientist
Nate Titterton

bstt
ANNUAL CONFERENCE
2014

2014-07-14
9am-noon
Broadway C

bjc.berkeley.edu
A Programming Approach to the CS Principles Data Task

**Agenda**

- **9:00-9:30 CSP Data Task**
- **9:30-9:45 What is Data?**
- **9:45-10:30 Small Data**
  - Intro to HOFs, *Snap!*, & all play with small dataset
- **10:30-10:45 Break**
- **10:45-11:30 Big(ger) Data**
  - Let’s work on a larger dataset together
- **11:30-11:45 Visualization**
  - Let’s connect
- **11:45-12:00 Summary**
How can we get computing into K-12?

- **New Course: “Computer Science : Principles”**
  - Engaging, accessible, inspiring, rigorous
  - Focused on the fundamental concepts of computing (Computational Thinking)
  - An impetus for college curriculum reform
  - Available nationwide (IB as well)

- **SINGLE SOURCE OF NATIONAL LEVERAGE!**

*CollegeBoard*

csprinciples.org
what is CS Principles?

6 computational thinking practices

- connecting computing
- developing computational artifacts
- abstracting
- analyzing problems and artifacts
- communicating
- collaborating

check out the complete curriculum framework at: csprinciples.org
what is CS Principles?

- computing is a **Creative** activity.
- **Abstraction** reduces information and detail to facilitate focus on relevant concepts.
- **Data** and information facilitate the creation of knowledge.
- Algorithms enable us to develop and express solutions to computational problems.

---

7 big ideas

- Programming enables problem solving, human expression, and creation of knowledge.
- The **Internet** pervades modern computing.
- Computing has global **Impacts**.

---

check out the complete curriculum framework at: csprinciples.org
Assessment of CS Principles Course

- **Old School (AP CS A)**
  - 3-hour written exam at end of year

- **Hybrid (AP CSP)**
  - 2 Performance Tasks throughout the year
    - Create “code” (pair & solo)
    - Explore “impact” (solo)
    - Investigate “data” (pair)
  - 2-hour computer-delivered exam at end of the year
3. Data and Information

1. People use computer programs to process information to gain insight and knowledge.
   1. Use computers to process information, find patterns, and test hypotheses about digitally processed information to gain insight and knowledge
   2. Collaborate when processing information to gain insight and knowledge
   3. Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notation, and precise language

2. Computing facilitates exploration and the discovery of connections in information.
   1. Extract information from data to discover and explain connections, patterns, or trends (Students are not expected to know specific formulas or options available in spreadsheet or database software packages)
   2. Use large data sets to explore and discover information and knowledge

3. There are trade-offs when representing information as digital data
   1. Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information
What about AP Statistics?

- Your students **will have been exposed** to data, graphs, & analyses in prior math classes
- **AP Stats** covers some of this material in depth

- How does **CS Principles** fit in?
  
  *Key: the computer*
Computers and Data

- *All* data analysis uses computers now
- *In the old days:*
  - Books of random numbers
  - Graphs designed for paper and pencil
- **CSP course is the link**
  - Trajectory of learning, early math → advanced stats
  - The field of Data Sciences
- **For non-CS students, the Data unit of CSP is arguably the most important to their lives!**
**Exploratory vs Confirmatory Analyses**

- **Exploratory:**
  - Describing what you see
  - Visualizing / Graphing your data
  - Less powerful making claims about underlying processes

- **Confirmatory**
  - **P-values!** A probability that what you observe isn’t just due to chance.
  - Wealth of mathematical discoveries underlay these statistical techniques

**Suggestion: leave anything confirmatory to Stats**
Experiments carefully control the generation of data, and will (generally) end with confirmatory analyses

- A/B testing is a form of experimentation used throughout the tech industry, especially in UI design

Observational studies look at data that already exist

- Most “interesting” datasets are of this type

Suggestion: generally avoid experimental data
“Big” Data

- A buzzword starting less than a decade ago, with many definitions
  - One definition, in industry: datasets that don’t fit on a single computer... terabytes!
- Good
  - Requires computers.
- Not good
  - Makes everything harder

Take home: use “big enough” data (1000+ rows)
A Workflow for Data Analysis

- **Find** your question, area of interest
  - Research your area
  - Design your study
  - Identify your data sources

- **Collect** your data, get it into your computer
  - Run the experiment
  - Scrape the web
  - Connect to public APIs

- **Clean** your data
  - Fix any errors

- **Analyze** your data
  - Appropriately visualize your data
  - Confirm or refine hypotheses, reanalyze
  - From the curriculum framework: “cluster, classify, find patterns”

- **Communicate** your results
Where should your focus be in CSP?

- **Find** your question, area of interest
  - Piloters responded that this was hard for many
- **Collect** your data, get it into your computer
  - Nice computational problems!
- **Clean** your data
  - Nice computational problems, can be tedious and boring
- **Analyze** your data
  - Nice computational problems
  - Don’t spend all your time here
- **Communicate** your results
  - Required by Curriculum Framework
UC Berkeley’s BJC
The Beauty and Joy of Computing

- 2009Fa: 16 students (pilot)
- 2010Fa: 90 students
- 2011Sp: 90 students
- 2011Su: ~25 HS teachers in BJC Family!
- 2011Fa: 250 Students
- 2012Sp: 250 Students
- 2012Su: ~100 HS teachers online!
- 2012Fa: 250 Students & 60 UCB online pilot
- 2013Sp: 250 Students
- 2013Su: ~175 HS teachers in BJC Family!
- 2013Fa: 360 Students
- 2014Sp: 250 Students
- 2014Su: ~250 HS teachers (~10 faculty) in BJC Family

Grant Winner
Pilot
Pilot x3
Award Winner

CS10.berkeley.edu
BJC curriculum and team leads

- Dan Garcia
  - Sr Lecturer SOE
  - Tiffany Barnes
    - NC State
- Brian Harvey
  - Sr Lecturer SOE
  - Nate Titterton
    - Research Scientist
    - ...many others @ Cal!
- Luke Segars
  - TA Grad Student
    - (Now @ Google)
- Colleen Lewis
  - TA Grad Student
    - (Now Prof @ Harvey Mudd)
BJC Initial High School Collaborators

Ray Pedersen
- Albany HS

Sean Morris
- Ralph Bunche HS

Eugene Lemon

Josh Paley
- Gunn HS

A Programming Approach to the CS Principles Data Task
What differentiates BJC?

- **More programming-centric**
  - Recursion
  - Functions-as-data, λ
  - Concurrency, Distributed computing

- **Graphic language makes difficult coding concepts easier**
  - Snap! (based on Scratch) is great

- **Unintended implications of computing technology**
  - Balanced optimism with pessimism
BYOB/Snap! add functions, generic lists, \lambda

- **BYOB (Build Your Own Blocks) \rightarrow Snap!**
  - developed by Jens Mönig w/design input and documentation from Brian Harvey & others @ Cal
  - Leverages awesomeness of Scratch (design, simplicity, multi-media, community of users)
  - Snap! is in Javascript, in-the-browser

Building a For Loop and calling it. What other languages make it this easy?
Let’s dive into Snap!

Activities. Everyone…

- Get on WiFi
- Load [http://tinyurl.com/BJCatCSTA2014Data](http://tinyurl.com/BJCatCSTA2014Data)
- Click on the upper-left cloud icon → Signup…
- Put in your information
- Check your email for the email from *Snap!* to log in
- Dan demos blocks, functions
Why use functions?

The power of generalization!

A Programming Approach to the CS Principles Data Task
But how general can we be?

The power of **generalization**!

A Programming Approach to the CS Principles Data Task
Higher-Order Functions

- Functions as Data
- Useful HOFs (you can build your own!)
  - **map** `Reporter` **over** `List`
    - Report a new list, every element `E` of `List` becoming `Reporter(E)`
  - **keep items such that** `Predicate` **from** `List`
    - Report a new list, keeping only elements `E` of `List` if `Predicate(E)`
  - **combine with** `Reporter` **over** `List`
    - Combine all the elements of `List` with `Reporter(E)`
    - This is also known as “reduce”
combine with **Reporter** over **List**
HOFs are one of our Big Ideas!

- Functions as data is one of the two (programming) big ideas in this course
- It’s a beautiful example of the abstraction of the list iteration details
- Google (and other companies) use this!
  - They use “map-reduce”
Small Data Explorations

- Let’s explore a very very small data set
- [http://snap.berkeley.edu/smalldata.txt](http://snap.berkeley.edu/smalldata.txt)

<table>
<thead>
<tr>
<th>Country</th>
<th>FIFA Titles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Germany</td>
<td>4</td>
</tr>
<tr>
<td>USA</td>
<td>0</td>
</tr>
<tr>
<td>Brazil</td>
<td>5</td>
</tr>
</tbody>
</table>
Peter Norvig’s classic “The Gettysburg Powerpoint Presentation”

A Programming Approach to the CS Principles Data Task
Big(ger) Data Explorations

- "Baby Names" dataset:
  - [www.ssa.gov/oact/babynames/limits.html](http://www.ssa.gov/oact/babynames/limits.html)
  - We stored 1880’s as: [snap.berkeley.edu/bigdata.txt](http://snap.berkeley.edu/bigdata.txt)
Visualization

- The “last mile” for data analysis
- Good fit for CSP
  - Best w/observational data
  - Showcases creativity
- Pitfalls
  - Subjective criteria
  - Young discipline
- Can CSP students feel engaged?
Visualization … Epic FAIL

Fox News Graphic on the “Obamacare” Enrollment as of 2014-03-27

A Programming Approach to the CS Principles Data Task
Visualization ... Epic WIN (1869)

Charles Joseph Minard, Napoleon’s 1812 Russian Campaign

A Programming Approach to the CS Principles Data Task
Visualization … Epic WIN (2009)
Our first attempts at integrating Snap with web services

- Goal: supporting real data and professional tools in Snap
- There are a wealth of directions to go with this and future integrations
- We need your feedback, what do you want?
Summary: Work in Data everywhere

- Working with lists or matricies?
  - Use a real dataset!
- Random?
  - Do a monte-carlo experiment!
- Creativity?
  - Discuss beautiful visualizations!
- Impacts?
  - Look at infographics!

Exposure to CS leads to some of the best-paying jobs in the world. But 75% of our population is underrepresented.
Resources (part 1)

- **Snap**: [snap.berkeley.edu](http://snap.berkeley.edu)
- **BJC**: [bjc.berkeley.edu](http://bjc.berkeley.edu)
- **BJC Data Lecture Video**: [youtu.be/cyVLW996TvY](https://youtu.be/cyVLW996TvY)
- **BJC Data Lecture Slides**: [https://inst.eecs.berkeley.edu/~cs10/sp14/Lectures/L18%20-%20Data/](https://inst.eecs.berkeley.edu/~cs10/sp14/Lectures/L18%20-%20Data/)
- **BJC Lab Curriculum (Topic 14 Data Unit)**: [http://bjc.berkeley.edu/bjc-r/course/cs10_sp14.html](http://bjc.berkeley.edu/bjc-r/course/cs10_sp14.html)
  - GPS “forensics” activity
  - Spam vs Ham detector
“An unusual episode”:
http://www.amstat.org/publications/jse/v3n3/datasets.dawson.html
- Usually done without computers; by providing raw data you can open up a computer-based task, however: http://tinyurl.com/BJC-UnfortunateEpisode

“Baby Names” dataset:

Data and Story Library
- http://lib.stat.cmu.edu/DASL

Spurious Correlations
- http://www.tylervigen.com/