AP CS PRINCIPLES

New AP Computer Science Class
Frank Hulsman, St. Xavier High School
SESSION AGENDA

• Introduction
• What is AP CS Principles?
• Why a new AP Class?
• Broaden Participation
• Timeline
• Course Curriculum

• Student Assessment
• AP CS Course Comparisons
• How do we get started?
• Teacher Training?
• Curriculum Resources
• Textbooks?
INTRODUCTION

• Presenter:
  • Frank Hulsman
  • 12 year Computer Science Teacher
  • 25 years experience in industry as computer programmer, systems analyst
  • St. Xavier High School – One of 40 High Schools selected to participate in the AP Computer Science Principles course “pilot”
  • Secretary - Kentuckiana Chapter of Computer Science Teachers Assoc (CSTA)
WHAT IS AP COMPUTER SCIENCE PRINCIPLES?

- New, widely accessible AP course for computer science.
- Official course website: apcsprinciples.org
- Designed to be far more than a traditional introduction to programming
- Rigorous, engaging, and approachable course
- Explores many of the big, foundational ideas of computing
- So that all students understand how these concepts are transforming the world we live in
WHY?

• To increase the number of students interested in and prepared for success in computer science and other STEM fields
• Funding from the National Science Foundation
OPPORTUNITY

1,000,000 more jobs than students by 2020

$500 billion opportunity

1.4 million computing jobs

400,000 computer science students
Computer science is the highest paid college degree and computer programming jobs are growing at 2x the national average.

- **Students**: 98% All other math and sciences: 2%

- **Jobs**: 60% All other math and sciences: 40%
BROADEN PARTICIPATION

• AP Computer Science Principles will give students the opportunity to:
  • Use current technologies to solve problems
  • Create meaningful computational artifacts

• Together, these aspects of the course make up a rigorous and rich curriculum that aims to broaden participation in computer science
KENTUCKY STATISTICS

• According to the college board, in 2013 *
  • 672 Kentucky students took AP Computer Science A Exam
  • 508 classified themselves as “white”
  • 96 Asian
  • 16 Black

• We need to increase and broaden participation in computer science!

* http://research.collegeboard.org/programs/ap/data/archived/2013
TIMELINE

• 2016 – 2017 Academic Year – Course Launch. First exam May 2017
• 2015 – 2016
  • 2014 – 2015 Three year pilot, 40 high schools, 10 colleges
  • 2013 – 2014
  • 2012 – 2013 Initial Pilot 10 high schools, 10 colleges
  • 2011 – 2012
• 2010 – 2011 Course started at 5 colleges / universities
COURSE CURRICULUM

• AP CS Principles offers a multidisciplinary approach to teaching the underlying principles of computation.

• Introduce students to:
  • Creative aspects of programming
  • Using abstractions and algorithms
  • Working with large data sets
  • Understandings of the Internet and issues of cybersecurity
  • Impacts of computing that affect different populations.
STUDENT ASSESSMENT

• Students taking the end of year assessment will receive a grade of 1-5
  • 40% of grade based on portfolio of work, uploaded to the college board
  • 60% of grade based on multiple choice exam administered in May

• Portfolio (Performance Tasks):
  • A written assignment focusing on a computing innovation and it’s impact
  • A programming project, consisting of an individually developed program and a collaboratively developed program
## AP CS COURSE COMPARISONS

<table>
<thead>
<tr>
<th>AP Computer Science A</th>
<th>AP Computer Science Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focused on object-oriented programming and problem solving</td>
<td>Built around fundamentals of computing including problem solving, working with data, understanding the internet, cyber security, and programming</td>
</tr>
<tr>
<td>Java programming language</td>
<td>Teachers choose the programming language(s)</td>
</tr>
<tr>
<td>Encourages skill development among students considering a career in computer science or other STEM fields</td>
<td>Encourages a broader participation in the study of computer science and other STEM fields, including AP Computer Science A</td>
</tr>
</tbody>
</table>
| Multiple-choice and free-response questions (written exam) | • Two performance tasks students complete during the course to demonstrate the skills they have developed  
• Multiple-choice questions (written exam) |
GETTING STARTED

• Add the course to your school curriculum
• Identify teacher(s)
• Teacher Training
• Recruit students
• Start your course either in the Fall of 2015 (non-AP), or plan to start in the Fall of 2016 (first year of AP)
TEACHER TRAINING

• During the summer of 2015, several groups will be offering teacher training
• More opportunities will be added in the summer of 2016
  • CS 4 Alabama – Recordings of last summer’s teacher training session available online at: https://csp-cs4hs.appspot.com
  • Code.org - In-person professional development for the Code.org CSP course will be available in select partner districts starting in summer 2015. In 2016, we will make all of our PD agendas, schedules, and other materials available to the public. This includes all of our online and in-person PD programs
  • Mobile CSP – Online Teacher Training: https://ram8647.appspot.com/teach_mobileCSP/course
CURRICULUM RESOURCES

• From the College Board website http://apcsprinciples.org/cs-principles-curricula/

• Ready-to-go, complete curricula that offer high school teachers plans, resources, tests, and more. Many teachers use parts of different curricula, but some prefer to use a single source for teaching.

• Mobile CSP: http://mobile-csp.org/

• Thriving in our Digital World: https://canvas.instructure.com/courses/884561

• Project Lead the way Project has a complete curriculum, available for PLTW schools part of the Computer Science program

• Code.org, available this summer: http://code.org/educate/csp
MORE CURRICULUM

• Online Courses / MOOCS
  • CS Principles, CS4HS Alabama https://csp-cs4hs.appspot.com
  • Mobile CSP https://ram8647.appspot.com/teach_mobileCSP

• This course may have more than enough material for a complete course, but may lack tests, may not have complete coverage of all learning objectives:
  • Beauty and Joy of Computing (NC State, UC Berkeley) http://bjc.berkeley.edu/website/curriculum.html

Or:
Dr. Tiffany Barnes http://bjc-nc.github.io/bjc-course/curriculum/
• There are not any official textbooks for this course. Most of the curriculum utilizes current websites.

• One free, online, book used by many courses: Blown to Bits, http://www.bitsbook.com/
• Additional Material – not covered during session
EXPLORING COMPUTER SCIENCE

• A beginner computer science curriculum
• Funded by National Science Foundation
• Year-long
• Curriculum available to any school: http://www.exploringcs.org/
# Computer Science Principles

## Unit 1 - Digital Information
**Overview:** This unit explores the technical challenges and questions that arise from the need to represent digital information in computers and transfer it between people and computational devices. Topics include: the digital representation of information - numbers, text, images, and communication protocols.

**Lessons:**
1. Impact of Innovation
2. Sending Binary messages
3. More complex messages
4. Bit Sending Widget
5. Sending Bits in the Real World
6. Number Systems
7. Binary numbers
8. Sending numbers
9. Encoding Numbers in the Real World
10. Encoding and Sending Text
11. Sending formatted Text
12. Bytes and File Sizes
13. Text Compression
14. Encoding B&W images
15. Encoding Color images
16. **Project** - Personal Favicon

### Practice PT: Encode a complex piece of information
- 30 days
- 6 weeks

### Practice PT: Internet and Privacy Issues
- 30 days
- ~6 weeks

### Practice PT: Writing about Code
- 40 days
- ~8 weeks

### Practice PTs: [integrated into projects]

### Real PTs
- 10-15 days
- ~2-3 weeks

(Example of instructional days each month: Chicago Public Schools 2015-16)

## Unit 2 - The Internet
**Overview:** Students continue to work together to invent solutions and protocols to many of the problems that arise in the structure and function of the Internet. Topics include the Internet Protocol, DNS, TCP/IP, cryptography and other security and hacking concerns.

**Lessons:**
1. Addressing Messages
2. Addressing for Computers
3. Name-to-address mapping
4. Finding your friends
5. IP Addresses, URLs, and DNS
6. Packet Switched Networks
7. Problems with packets
8. Redundancy and Routing
9. Who can see my data?
10. Routing, Scaling, The Internet
11. Securing our bits
12. intro to encryption
13. Computationally hard problems
14. “Hard” security
15. Public Key Cryptography
16. Security and Hacking in the real world
17. intro to AppLab
18. Putting information on the web

### Practice PT: Encode a complex piece of information
- 30 days
- 6 weeks

### Practice PT: Internet and Privacy Issues
- 30 days
- ~6 weeks

### Practice PT: Writing about Code
- 40 days
- ~8 weeks

### Practice PTs: [integrated into projects]

### Real PTs
- 10-15 days
- ~2-3 weeks

(Example of instructional days each month: Chicago Public Schools 2015-16)

## Unit 3 - Programming
**Overview:** Students learn the fundamentals of programming constructs of JavaScript by solving problems with “turtle graphics” and then simple event-driven apps. An emphasis is placed on enabling students to solve problems and write code in teams.

**Lessons:**
1. Solving Big Problems with CSS
2. The Need for Programming Languages
3. Quick Tour of Pencil Code
4. Programming with Primitives
5. Function Junction
6. Function Challenge
7. Parameterized Functions
8. **Project**: Design Your Own Font
9. Application Programming Interface
10. Team Design
11. Team Programming
12. Variables
13. Variable Arithmetic
14. Numeric input
15. Interactive Apps with Variables
16. Conditional (to be re-written)
17. If Statements (to be re-written)
18. Nested if-statements
19. While Loops (to be written)
20. **Project**: Quiz App
21. What does the Internet know about you?
22. Extracting knowledge from data

### Practice PT: Encode a complex piece of information
- 30 days
- 6 weeks

### Practice PT: Internet and Privacy Issues
- 30 days
- ~6 weeks

### Practice PT: Writing about Code
- 40 days
- ~8 weeks

### Practice PTs: [integrated into projects]

### Real PTs
- 10-15 days
- ~2-3 weeks

(Example of instructional days each month: Chicago Public Schools 2015-16)

## Unit 4 - Data
**Overview:** Students continue programming and building web apps, now with a view towards data: processing it with algorithms, and the implications of storing data collected from and by others on the web. Students explore “big data” by making queries against a data analysis api.

**Lessons:**
- Utility of Random numbers
- A coin flipping experiment
- Construct your own experiment
- **Project**: Run your experiment
- Processing Lists and what “Big Data” means
- Process a small list
- Manipulate items in a list
- Investigate image data
- Simple image filters
- **Project**: Make your own image filter
- Storing user data in a database
- Retrieving data from your database
- The risks of storing personal data
- **Project**: Make an app that uses data from a database
- Experiments with big data
- Programming against an api (wolfram)
- Extracting knowledge from data

### Practice PT: Encode a complex piece of information
- 30 days
- 6 weeks

### Practice PT: Internet and Privacy Issues
- 30 days
- ~6 weeks

### Practice PT: Writing about Code
- 40 days
- ~8 weeks

### Practice PTs: [integrated into projects]

### Real PTs
- 10-15 days
- ~2-3 weeks

(Example of instructional days each month: Chicago Public Schools 2015-16)

## Unit 5 - Explore and Create
**Overview:** This unit is primarily dedicated time for students to prepare and finish their performance task projects. A few instructional days will allow the teacher to prepare the students before “letting them go” to work on their projects.

**Lessons:**
- Prepare for Create PT
- Do CREATE
- Prepare for Explore PT
- Do Explore

### Practice PT: Encode a complex piece of information
- 30 days
- 6 weeks

### Practice PT: Internet and Privacy Issues
- 30 days
- ~6 weeks

### Practice PT: Writing about Code
- 40 days
- ~8 weeks

### Practice PTs: [integrated into projects]

### Real PTs
- 10-15 days
- ~2-3 weeks

(Example of instructional days each month: Chicago Public Schools 2015-16)