Computer Science Teachers
1. What should be the goals of K-12 Computer Science education? At the end of computer science students should....

CS students, grade 12, should be able to understand and code in one universally acknowledged programing language (HTLM, C - Various, Java, or Python).

Students should be able to take those skill sets and create a simple program.
### 2. What content should be covered in K2 computer science education?

<table>
<thead>
<tr>
<th>Cognitive Process</th>
<th>K-2</th>
<th>3-5</th>
<th>6-8</th>
<th>9-10</th>
<th>11-12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Algorithms</td>
<td>Functions</td>
<td>Debugging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>When the Course should be taught</td>
<td>Afterschool Enrichment</td>
<td>Enrichment</td>
<td>Course offered as an Electives MYCS</td>
<td>Course offered as an Elective: CS 1, CS 2</td>
<td>Course offered as an Elective: CS 3, CS 4 - AP</td>
</tr>
</tbody>
</table>
Current Computer Science Reference Guide

July 2016 - Update

Key Strands:
- Algorithms and Programs
- Computing Systems
- Data and Analysis
- Networks and the Internet
Computational Thinking

- Sequencing and Order
- Use manipulatives

Algorithmic Thinking
3-5

Computational Thinking
  • Sequencing and Order
  • Use manipulatives

Algorithmic Thinking – Vocabulary Development
Computational Thinking

- Sequencing and Order
- Use manipulatives

Algorithmic Thinking
(Loops, Functions, Pattern Matching, Decomposing)

Vocab Development and Reinforcement
Intro to Programming Language

Course 3
Course 3 is a follow-up to Course 2.
Ages 8-18

Course 4
Students taking Course 4 should have already taken Courses 2 and 3.
Ages 9-18

Accelerated Course
Learn basic computer science in an accelerated version of courses 2-4.
Ages 10-18

6-8
Computational Thinking
- Sequencing and Order
- Use manipulatives

Algorithmic Thinking
(Loops, Functions, Pattern Matching, Decomposing)

Vocab Development and Reinforcement
Intro to Programming Language

### Organization | Curriculum | Professional Development
---|---|---
Beauty and Joy of Computing | Year-long CS Principles course, FREE | In-person in NYC, Berkeley, CA and North Carolina, FREE stipends in NYC, stipends for level elsewhere paid as available
Bootstrap | Teach algebra through video-game programming, with a 10-yr module to go alongside or inside a math class | 3-day workshops for schools and districts. Fees range
CodeHS | 4-year high school CS pathway. Intro CS, JavaScript, Intro CS Python, AP CS Principles, AP CS in Java, Computing Ideas, Web Design and more, FREE. Pro plans for schools start at $2500 | Online PD for Teaching Intro CS, Teaching AP Java, and Teaching AP CS Principles, 30-40 hour course, $1500/teacher
Edhesive | Year-long AP Computer Science course, FREE | Online PD, on-campus and content/teacher support available to the school
Exploring Computer Science | Year-long introductory high school course aimed at broadening participation in CS. 6 units, 6 weeks each | Week-long summer institute and quarterly one-day academic year workshops
Globalloria | 6 game-design courses, $75/student | 5-day in-person training

### Course 3
Course 3 is a follow-up to Course 2. Ages 8-18

### Course 4
Students taking Course 4 should have already taken Courses 2 and 3. Ages 9-18

### Accelerated Course
Learn basic computer science in an accelerated version of courses 2-4. Ages 10-18

### Code Combat
Forever imagine, program, share
Ready for the next step?
JavaScript Tools for High School

**App Lab**
Ages 13+
App Lab is a programming environment where you can make simple apps. Design an app, code with blocks or JavaScript to make it work, then share your app in seconds.

**Computational Thinking**
- Sequencing and Order
- Use manipulatives

**Algorithmic Thinking**
(Loops, Functions, Pattern Matching, Decomposing)

**Vocab Development and Reinforcement**

**Programming Language**

**AP - Courses**
Knowledge and Capabilities would define a College-Ready CS Student
Knowledge and Capabilities would define a Career-Ready CS Student.
# 3. Scope and Sequence Middle School

<table>
<thead>
<tr>
<th>Month</th>
<th>Topic</th>
<th>Links to Items</th>
<th>Topic</th>
<th>Links to Items</th>
</tr>
</thead>
</table>
| AUG   | 15 instructional days | - Light Bot (First 2 Weeks of School Year)  
- HMC (What makes a Computer) | DEC | 12 instructional days  
- Google CS first Model (Hackathon) |
| SEPT  | 16 instructional days  
- Labor Day 5th  
- PD 20th | - Scratch Basics  
- Basic Blocks  
- Stage Specifications (X and Y axis)  
- Debugging activity  
- Animate Your Name Activity  
- Video Game 1 (Basket Catch)  
- Variables, and If, Else  
- Broadcasting  
- Video Game 2 (Basic Maze - Color Sensing)  
- (TV shows: Limitless Sequence showing Mind Maps)  
  - Video Game 1 (Basket Game)  
  - Variables and Broadcasting  
  - Color Sensing  
- Lunch Time Arcade | JAN | 21 Instructional Days  
- MLK 16th  
- App Builder  
http://appinventor.mit.edu/explorer/sites/all/files/ConceptCards/al2/AppinventorMakerCards.pdf  
Or APP LAB  
https://code.org/educate/applab |
| OCT   | 21 Instructional Days  
- CODE.ORG Course 3  
For early finishers (Challenge course 4) (Review course 2) | FEB | 17 Instructional Days  
- Lincoln - 13  
Washing -20 |
| NOV   | 16 Instructional Days  
- Vet Day 11th  
PD day NOV 1st  
Thanksgiving | MARCH | 20 Instructional Days  
- Basics of robots  
- Finch Bots - Great Maze Challenge | |
|       |       | Video Productions  
- Based off the an article or literature (Greek myths or Shakespeare) or a Cause and Effect (Microbead) have students complete and create (*Mini Video)  
- Looking at a Song with Strong idiomatic language. Create a video the best reflects the ideas | APRIL | 15 Instructional Days  
- Finch Bots - Great Maze Challenge |
|       |       | Story Teller  
- Info-graphic that shows how the student view the Computer Science profession  
- Think idiomatically | MAY/JUNE | 24 Instructional Days  
- Foundational Python for Kids |
|       |       | CODE Academy - Python for Kids |
4. California CSS connection to CSS and other content areas
5. Other Computer Science Standards will be useful tool for California Educators?