El Monte Union High School District

4/20/16

Course Outline

High School DISTRICT

Title: AP Computer Science Principles

Transitional*_____(Eng. Dept. Only)

Sheltered (SDAIE)*____Bilingual*___

AP**__X_____Honors**_____

Department: Business (CTE)

Grade Level (s):_____9-12_____

Semester_____Year__X_____

Year of State Framework Adoption 2016

This course meets graduation requirements:

( ) English
( ) Fine Arts
( ) Foreign Language
( ) Health & Safety
( ) Math
( ) Physical Education
( ) Science
( ) Social Science
(X) Elective

Department/Cluster Approval	Date
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

*Instructional materials appropriate for English Language Learners are required.

**For AP/Honors course attach a page describing how this course is above and beyond a regular course. Also, explain why this course is the equivalent of a college level class.

AP Computer Science Principles (APCSP) is a full year, rigorous, entry-level course that introduces high school students to the foundations of modern computing. The course covers a broad range of foundational topics such as programming, algorithms, the Internet, big data, digital privacy and security, and the societal impacts of computing.

1. Prerequisite(s):

Algebra 1 or Integrated Math 1

2. Short description of course which may also be used in the registration manual:

The course introduces students to the foundational concepts of computer science and challenges them to explore how computing and technology can impact the world. The AP Program designed AP Computer

Science Principles with the goal of creating leaders in computer science fields and attracting and engaging those who are traditionally underrepresented with essential computing tools and multidisciplinary opportunities.

3. Describe how this course integrates the schools ESLRs (Expected School-wide Learning Results):

All AP® Computer Science Principles (APCSP) students are expected to apply critical thinking skills and problem solving skills as they complete computer problems and projects. (*Apply Academic Skills.)* They demonstrate their computation skills by working on various programming projects. Technology such as computers are used to code and solve problems. *(Utilize technology as a tool. Apply career readiness skill.)* As students work in groups and teamwork is fostered to reinforce interpersonal skills. *(Apply Personal & Interpersonal Skills.)* Since teams are formed among various ethnic groups, students learn to work in a culturally diverse environment. *(Show respect for diversity.)* Personal skills are practiced by students as all assignments are expected to be completed in a timely manner and of the highest quality.

4. Describe the additional efforts/teaching techniques/methodology to be used to meet the needs of English Language Learners:

Students are instructed using various techniques such as small group discussions, mini lectures, computer presentations, demonstrations, YouTube videos, and one-on-one tutoring.

5. Describe the interdepartmental articulation process for this course:

Since the objectives of APCSP expect students to read and write code, communicate, calculate, and document their work, students must apply their knowledge in language arts and math to complete assignments and projects. In addition, students are exposed to math concepts as they learn about computer programming.

6. Describe how this course will integrate academic and vocational concepts, possibly through connecting activities. Describe how this course will address work-based learning/school to career concepts:

This class combines computer concepts with hands-on activities and simulation. Therefore, students are expected to integrate their academic knowledge with vocational principles to complete assignments and projects. Through the course of the year, students are able to make connection between what they learn in core areas and translate these knowledge to real life scenarios in career technical education classes, which would provide the relevancy for mastering core subjects.

7. Materials of Instruction (Note: Materials of instruction for English Language Learners are required and should be listed below.)

A. Textbook(s) and Core Reading(s):

Reference Text

Note: Students do not have their own copy of this book
MacCormick, John. *Nine Algorithms that Changed the Future: The Ingenious Ideas that Drive Today's Computers*. Princeton, NJ: Princeton University Press, 2012.

Blown to Bits http://www.bitsbook.com/

This course does not require or follow a textbook. Blown to Bits is a book that can be accessed online free of cost. Many of its chapters are excellent supplemental reading for our course, especially for material in Units 1, 2 and 4. We refer to chapters as supplemental reading in lesson plans as appropriate.

Programming Environments

· Code.org
· "Scratch," *MIT Media Lab* at scratch.mit.edu.
· "Snap!," *University of California, Berkeley* at snap.berkeley.edu.
· Python.org.
· "CodeSkulptor," *Scott Rixner* at codeskulptor.org.

B. Supplemental Materials and Resources:

Online Resources

Online articles and videos from sources such as *New York Times* (nytimes.com), Wikipedia (wikipedia.org), Logicly (logic.ly), and YouTube (youtube.com).

C. Tools, Equipment, Technology, Manipulatives, Audio-Visual:

Computers, Overhead Projector, ELMO, Insight,

8. Objectives of Course and Student Performance Standards

- **Objectives of Course**

  The goal of AP Computer Science Principles is to provide a broad, inspiring overview of computer science that is appropriate for all students who have completed a high school algebra course. By the end of this course, students will become empowered to critically analyze computing innovations as well as create inspiring applications that express their interests. In addition, they will be ready to incorporate computational thinking into their future fields of study.

- **Unit detail including projects and activities including duration of units (pacing plan)**

  ### Course Overview

The units that follow interweave the six Computer Science Principles Computational Thinking Practices listed below:

· P1: Connecting Computing
· P2: Creating Computational Artifacts
· P3: Abstracting
· P4: Analyzing Problems and Artifacts
· P5: Communicating (both orally and written)
· P6: Collaborating

Along with the seven Computer Science Principles Big Ideas:

· Big Idea 1: Creativity
· Big Idea 2: Abstraction
· Big Idea 3: Data and Information
· Big Idea 4: Algorithms
· Big Idea 5: Programming
· Big Idea 6: The Internet
· Big Idea 7: Global Impact

Assessment

Students are primarily evaluated on the basis of their work, which can take the form of worksheets, writing assignments, programs, and online journal entries. From time to time, quizzes are given which check for understanding of essential skills and knowledge.

Units

Unit 1: Introduction to CS Principles (Creativity, Algorithms, Global Impact)

Guiding Questions

· How does continuous access to large amounts of data change how people and organizations make decisions?
· How do computers put things in order and find things in a list?
· What is the connection between data, information, knowledge, and wisdom?

Lessons

· Impact on your life
· What is an algorithm?
· What is a program?
· Program or be programmed

· Experiments on social media users
· Programming as a form of expression
· Making music and art in Scratch

Instructional Activity: Impact on Your Life

On the first day of class, I ask students, "What computing innovation has had the most impact on your life?" Students consider the question individually, in small groups, and as a class. That night they have a conversation with an adult in their life and report back that person's answer. The next day students are asked to write a document that includes both their response and the adult's response. On the third day, I present the seven CSP big ideas to students. Students complete a chart that provides an example of each big idea as it relates to the innovation they chose. LO 7.1.1[P4] **[CR1a] [CR2g]**
[CR1a] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P1: Connecting Computing.
[CR2g] — Students are provided with opportunities to meet learning objectives within Big Idea 7: Global Impact. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.

Unit 2: The Internet (Data and Information, Algorithms, Programming, The Internet, Global Impact)

Guiding Questions

· How does continuous access to large amounts of data change how people and organizations make decisions?
· How do computers put things in order and find things in a list?
· What is the connection between data, information, knowledge, and wisdom?

Lessons

· Web crawlers
· Indexing pages
· Ranking pages
· Privacy on the web
· Internet origins and governance
· Simulating TCP/IP
· Domain name servers
· Coding with if statements
· Scaling and net neutrality
· Cybersecurity in the news
· Acting out a DDoS attack
· Scratch project: The Internet
· Privacy and the government
· Is the Internet broken?

Instructional Activity: Domain Name Servers

Students go through a collaborative activity that demonstrates how domain name servers work. One student acts as the Domain Name Server, and the other students act as individual internet users. The individual users write a domain name (such as collegeboard.org) on a piece of paper and pass it to the person who is the server. That person turns the domain name into an IP Address (such as 128.23.01.22). The class pauses and discusses how this might actually happen on the Internet, and how a single system could scale to handle the large number of domain names and users on the Internet. Finally, students visit a website that shows how domain name servers work. LO 6.1.1[P3], LO 6.2.1[P5], LO 6.2.2[P4] **[CR1c] [CR1d] [CR1e] [CR2f]**
[CR1c] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P3: Abstracting.
[CR1d] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P4: Analyzing Problems and Artifacts.
[CR1e] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P5: Communicating (both orally and written).
[CR2f] — Students are provided with opportunities to meet learning objectives within Big Idea 6: The Internet. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.

Unit 3: Artificial Intelligence (Creativity, Abstraction, Algorithms, Programming, Global Impact)

Guiding Questions

· How do computers act in intelligent ways?
· How do we define artificial intelligence?
· How can algorithms be written to win games?
· How have competitions between humans and computers defined what intelligence means?

Lessons

· What is AI?
· The Turing test
· Modern Turing tests
· History of AI
· Machine learning
· Natural language processing
· Are algorithms taking over?
· Programming decision trees
· Modulo arithmetic
· Programming the game of stones
· Programming the game of rock-paper-scissors
· IBM's Deep Blue
· IBM's Watson

· Final jeopardy betting algorithm
· Humans and robots
· Social intelligence
· Programming "today is…"

Instructional Activity: Rock-Paper-Scissors Against the Computer

Students play the game rock-paper-scissors against an online computer and discuss the computer's choosing algorithm. Then they learn a more advanced version of the game called "Rock, Paper, Scissors, Lizard, Spock" and play it in class against each other. As a whole class, students write an algorithm that represents the five words as integers and order the choices in such a way that a single line of code can determine the winner. This will help them learn how to think like a computer scientist. The next day, students implement their algorithm as a Scratch program. LO 2.2.1[P2], LO 4.1.1[P2], LO 4.1.2[P5], LO 5.1.2[P2], LO 5.1.3[P6] **[CR1b] [CR1e] [CR1f] [CR2b] [CR2d] [CR2e]**
[CR1b] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P2: Creating Computational Artifacts.
[CR1e] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P5: Communicating (both orally and written).
[CR1f] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P6: Collaborating.
[CR2b] — Students are provided with opportunities to meet learning objectives within Big Idea 2: Abstraction. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
[CR2d] — Students are provided with opportunities to meet learning objectives within Big Idea 4: Algorithms. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
[CR2e] — Students are provided with opportunities to meet learning objectives within Big Idea 5: Programming. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.

Unit 4: Abstraction and Simulation (Abstraction, Programming, The Internet)

Guiding Questions

· How can you convert numbers between binary, decimal, and hexadecimal form?
· What are some uses for binary and hexadecimal representations of numbers?
· How do computers add numbers, and how do they store numbers?
· How is a computer's random-access memory (RAM) and central processing unit (CPU) organized?
· How do random numbers allow computers to simulate real-world events?

Lessons

· Binary representation of data
· Reading about information theory
· Self-correcting codes
· Introduction to Snap!
· Hexadecimal numbers
· Review of "and, or, not,"
· Designing adders in Logicly
· Storing a bit in Logicly
· Simulating operation of the CPU
· Simulations and models
· Programming simulations with dice and coins
· Programming Monte Carlo simulations
· Programming the cereal box problem
· Simulating real-world events

Instructional Activity: Designing Adders in Logicly

Students use logic gates to demonstrate how computers add two bits. Logicly allows students to abstract a complicated circuit into a "box" with inputs and outputs. Once students create the "box" for an adder with carry in and carry out bits, they can build a circuit that adds two four-bit numbers. Students naturally see a problem when some results are not correct, and this leads to a student-constructed conversation about overflow error. Students write a report describing how abstraction hides levels of complexity. LO 2.1.2[P5], LO 2.2.1[P2], LO 2.2.3[P3] **[CR1b] [CR1c] [CR2b]**
[CR1b] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P2: Creating Computational Artifacts.
[CR1c] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P3: Abstracting.
[CR2b] — Students are provided with opportunities to meet learning objectives within Big Idea 2: Abstraction. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.

Unit 5: Data (Creativity, Abstraction, Data and Information, Algorithms, Programming, Global Impact)

Guiding Questions

· How does continuous access to large amounts of data change how people and organizations make decisions?
· How do computers put things in order and find things in a list?
· What is the connection between data, information, knowledge, and wisdom?

Lessons

· Innovations from data
· Big data
· Visualizing big data
· You-sort
· Sorting algorithms
· Coding bubble sort
· Binary search worksheet
· Programming a reverse guessing game
· Bioinformatics algorithms
· Lossless compression
· Lossy compression
· Coding data compression
· The data, information, knowledge, wisdom (DIKW) pyramid
· Gapminder.org
· Data use in your school
· Privacy in the age of big data
· Downloading public data into spreadsheets
· Manipulating data in Python

Instructional Activity: Applications from Data

The last 20 years have seen waves of trends in computing. Whether it was hardware, software, the Internet, search, social, or mobile, each wave created incredible consumer innovations as well as profits for companies that created those innovations. Will data be the next wave? Students play an online guessing game that is powered by crowd-sourced data, analyze the game, and collaborate by adding more information to the game's data. LO 1.2.5[P4], LO 3.2.2[P3], LO 7.2.1[P1] **[CR1a] [CR1c] [CR1d] [CR2a] [CR2c] [CR2f]**
[CR1a] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P1: Connecting Computing.
[CR1c] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P3: Abstracting.
[CR1d] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P4: Analyzing Problems and Artifacts.
[CR2a] — Students are provided with opportunities to meet learning objectives within Big Idea 1: Creativity. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
[CR2c] — Students are provided with opportunities to meet learning objectives within Big Idea 3: Data and Information. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
[CR2f] — Students are provided with opportunities to meet learning objectives within Big Idea 6: The Internet. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.

Unit 6: Intractable Problems (Algorithms, Programming, The Internet)

Guiding Questions

· What kinds of problems are hard for computers to solve?
· What kinds of problems are impossible for computers to solve?
· How do hard problems form the basis for modern encryption?

Lessons

· Intro to intractable problems
· Heuristics
· Cryptography
· Programming Caesar cipher in Python
· Public key encryption
· Certificate authorities
· Unsolvable problems

Instructional Activity: Cryptography

Students play a game called the Encryption Game. In this game, students communicate and collaborate in pairs to solve the problem of encoding a two-digit number. One student sends the message across the room so the entire class can see or hear it. The other students try to guess the encrypted number. Finally, the partner decodes the number. The class tries to determine the system. LO 6.3.1 [P1] **[CR1a] [CR2f]**
[CR1a] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P1: Connecting Computing.
[CR2f] — Students are provided with opportunities to meet learning objectives within Big Idea 6: The Internet. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
Performance Task: *Create—Applications from Ideas*
After completing Unit 6, students complete through-course assessment *Create—Applications from Ideas* (12 hours in class). **[CR4]**
[CR4] — Students are provided the required amount of class time to complete the AP Through-Course Assessment *Create - Applications from Ideas* Performance Task.

Unit 7: Global Impact (Global Impact)

Lessons

· Defining global impact
· Copyright and the law

· Class debates:
¸ Smarter or not smarter?
¸ Narrowing or widening inequity?
¸ Stronger or weaker relationships?

Instructional Activity: Global Impact

Students discuss the following question: What does it mean to have a large impact? What kinds of past innovations have had the most impact? I guide students through some previous global innovations such as the telephone. For each impact discussed, students fill out a worksheet listing the following: creativity, abstraction, data, algorithm, networking, beneficial and harmful effects, and impact on society. Students identify and evaluate credible sources of information in preparation for class debates. LO 7.3.1[P4], LO 7.4.1[P1], LO 7.5.1[P1], LO 7.5.2[P5] **[CR1a] [CR1d] [CR2g]**
[CR1a] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P1: Connecting Computing.
[CR1d] — Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P4: Analyzing Problems and Artifacts.
[CR2g] — Students are provided with opportunities to meet learning objectives within Big Idea 7: Global Impact. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
Performance Task: *Explore—Impact of Computing Innovations*
After completing Unit 7, students complete through-course assessment *Explore—Impact of Computing Innovations* (8 hours in class). **[CR3]**
[CR3] — Students are provided the required amount of class time to complete the AP Through-Course Assessment *Explore - Impact of Computing Innovations* Performance Task.

- **Indicate references to state framework(s)/standards (If state standard is not applicable then national standard should be used)**

  AP Computer Science Curriculum Framework

- **Student performance standards**

  See Above in Units 1-7

- **Evaluation/assessment/rubrics**

  The course provides a number of assessment types and opportunities. For students, the goal of the assessments is to prepare them for the AP exam and performance tasks. For teachers, the goal is to use assessments to help guide instruction, give feedback to students, and make choices about what to emphasize in lessons. Summative Assessments: The curriculum

contains two types of summative assessments that teachers may elect to use. They are intended to mimic the AP assessments though in more bitesized chunks.

1. Fixed Response Assessments
After a group of concepts has been adequately covered typically this means every 58 lessons (roughly every few weeks) a fixed response assessment with items such as multiple choice, matching, choose two, short answer, etc. appears in the curriculum.

2. Practice Performance Task Assessments
Each unit contains at least one project designed in the spirit of the Advanced Placement Performance Tasks (PTs). These Practice PTs are smaller in scope, contextualized to the unit of study and are intended to help prepare students to engage in the official administration of the AP PTs at the end of the course.

3. Project Rubrics
The curriculum contains rubrics for assessing certain kinds of student work:

- Written and project work
- Practice PTs
- Programming projects
- Student presentations

Formative Assessments:
The curriculum provides teachers many opportunities for formative assessment (such as checks for understanding). These include, but are not limited to:

Assessments in Code Studio
All lesson materials can be accessed by students on a single platform called Code Studio. In addition to housing lesson descriptions, instructional materials, and programming exercises in App Lab, Code Studio includes features that assist the teacher in completing formative assessment including:

- Multiple choice or matching questions related to questions on the chapter summative assessment.
- Freeresponse text fields where students may input their answer.
- Access to student work within the App Lab programming environment and other digital tools and widgets used in the curriculum.
- The ability for students to submit final versions of App Lab projects

Worksheets and Activity Guides

- Many lessons contain worksheets or activity guides that ask students to write, answer questions, and respond to prompts (Answer keys provided) that could be used as formative assessment

It is up to the classroom teacher:

- to determine the appropriateness of the assessments for their classrooms
- to decide how to use, or not to use, the assessments for grading purposes. The curriculum and Code Studio does not provide teachers with a gradebook, and we do not provide recommendations for how to assign grades based on performance on an assessment.

Include minimal attainment for student to pass course

Students must complete and pass 70% of all Performance Tasks and Tests