

AP Computer Science Principles Summer 2019 Assignment

You must have all the assignments below submitted NO LATER THAN August 25th. All of this is for points. You will not be in a position to earn an A in this course if you do not submit any of these assignments on time and to my satisfaction. No excuse for lateness will be valid.

Administrative/Logistics Stuff

1. Use your school email account to join my AP CSP Summer Assignment Google Classroom.
The “join code” is f37wohx
2. Email me with any questions at tambara.kevin@tUSD.org
3. There is no textbook for this course. Scratch help is excellent (just right click on any block), and there are lots of online tutorials and examples.
4. If you don't already have one, create a personal Scratch account at <https://scratch.mit.edu>
5. There will be a test the first week to assess your level of programming skills. As long as you complete the coding assignments on your own and understand what you did, you'll be fine.
6. Use the following link to access the AP CS Principles Course Description. There are many concepts and useful vocabulary words used in context. <https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf?course=ap-computer-science-principles>

Assignment 1 - algorithms and abstractions

1. In a few sentences, and in your own words, explain what an algorithm is. Write a 10 line minimum pseudo-code (regular English phrases; doesn't have to be complete sentences, bullet points) algorithm for one of the following: brushing your teeth, baking cookies, or some other common task familiar to you. For example, making ramen noodles:
 - 1) Fill pot with water
 - 2) Turn stove burner on high
 - 3) Open ramen noodle package
 - 4) etc.
 - .
 - .
 - .
 - 10) pick up utensil and start eating
2. In a few sentences, and in your own words, explain what an abstraction is. How do abstractions simplify our daily lives? For example, if your Mom or Dad asked you to “wash the dishes”, would they have to tell you exactly what to do, or would already know the “dish-washing” steps?
3. Why are the Scratch “broadcast” and “when I receive broadcast” blocks considered abstractions?
4. Submit this assignment as a Google Doc file to Google Classroom Assignment #1

Assignment 2 - Scratch programming (algorithms and abstractions)

1. Create a Scratch project that has a Sprite demonstrating the algorithm you wrote in assignment #1 by saying and doing each of the 10 steps. Your code should be activated by either the “green flag” or “space bar” event handlers. You may want to add other sprites/images to help support your Sprite algorithm demonstration.

Now add another Sprite to your Scratch project, and use the “broadcast message” and “when I receive message” blocks for the two Sprites to communicate with each other, and to illustrate your algorithm and abstraction.

2. “Share” your project (click share button), then copy and “add” the project link to Google Classroom assignment #2. Caution - you must “share” your project before anyone else can view it.

Assignment 3 - Scratch programming (user input and variables)

1. Create a Scratch project which prompts the user (Sensing “ask/wait” and “answer” blocks) for your name and says (Looks “says” block) “Hello” + name. You’ll want to use an Operator “join” block.
2. Now modify your project so immediately after it loops (“repeat” block) so that a Sprite counts (“say” block) from 1 to 20, and then counts backwards from 20 to 1.
3. Modify your program again to include user input (“ask/wait” and “answer” blocks) which allows the user to select any number to count forwards and then backwards.
4. “Share” your project (click share button), then copy and “add” the project link to Google Classroom assignment #3. Reminder - you must “share” your project before anyone else to view it.

Optional assignment (if you have time - ha ha)

1. We will learn most of our programming concepts and techniques through Scratch, and then apply them to the other languages we’ll be learning.
2. Try writing Scratch programs that involve “if-then-else” and “repeat/repeat until” blocks. These are called conditional blocks and control the flow, direction, and ultimate result of a program.
3. We’ll explore the impact of the Internet on society, and discuss some of the more relevant issues. Read Chapter 1 of “Blown 2 Bits” (B2B), found on Google Classroom.

**Computer science is NOT HARD,
but it is HARD WORK.**