

Q: Why is APCSP being offered?

A: Monta Vista has offered programming courses for a long time, but computer science is so much more than programming! Digital technology (and other technology) is central to who we are, how we make decisions, and how modern careers work. It is ideal for all students to learn about computer science in a broad way with clear connections to their worlds.

Q: How does this differ from Java Programming and AP CSA?

A: Java Programming is primarily a programming class using the Java language. APCSP is a broad computer science course teaching more than programming. APCSP does not use only Java, and syntax (how a language is written to have meaning) is not a heavy part of APCSP. AP CSA is an AP Programming course using the Java language, extending what is used in Java Programming

Q: Can I sign up without any programming experience?

A: Please do! There will be a lot of supports built into class for people with no programming experience. We'll have a lot of supports in place for people with no formal design experience, too! EVERYONE will be learning new languages in this course (we rarely use Java), so everyone will have time to learn syntax and advantages/disadvantages of different languages together.

Q: Can I do projects independently? (I don't like to collaborate / do group work.)

A: No. If you don't like to collaborate or do group work, you should not sign up for this class.

Q: I have taken AP CSA. Can I still take APCSP?

A: You can try, but priority will be given to those without AP CSA.

Q: What computer languages and tools are used in APCSP?

A: A whole bunch! Instead of creating digital projects to fit a specific language and its features (what is done in AP CSA and Java Programming), we will often pick a product, goal, problem, or target audience, and seek out tools and languages that work best with each. This is a design-first approach. For instance, we'll be using ontology generation tools when designing an information database. We'll use drag-and-drop object oriented tools to work with graphical elements. Often the specific language will be hidden by an interface – remember, this class is syntax light! Other projects include using proprietary languages for specific mobile tools, designing software that makes sense for *that* mobile tech, using interpreted languages for cloud-based analysis of big text data and big geographic data, and using object oriented languages to work with the details of music design (yes, music). We do a lot with art, music, and literature, in order to better understand HOW computers make logical decisions and WHY they are very limited when working with aesthetic information.

Q: How much homework will be assigned?

A: In 2016-17, students reported spending 30 minutes per day, 4 times per week, on average. There will be reading assignments, paper/product analysis, and some content review assignments, but as much of the course is collaborative, there may not be a lot of homework when collaboration is taking place; most of that work should be done in class. Readings mainly come from a Stanford-published Logic textbook, [Blown to Bits](#) (a non-fiction discussion of ethical, legal, and cultural shifts with technology), and current magazines. We do not use a textbook.

Q: What are key topics in the course?

A: Design principles (for understanding how tech works and building our own tech), Abstraction, Data and Information, Global Impact (societal impact, shifts in culture, business, law, security/privacy, etc.), Internet, Programming, Algorithms, Logic

Q: Who should I talk to if I have further questions?

A: Mrs. Frazier (email is great: debbie_frazier@fuhsd.org, or in person in A103 or B209)! You can check out what the College Board has to say at <https://advancesinap.collegeboard.org/stem/computer-science-principles>.