

AP Computer Science Principles Canyon Crest Academy

Instructor: Michael Remington
michael.remington@sduhsd.net
858-350-0253 ext.4176

Location: F101/F106

Course Description:

The AP Computer Science Principles course provides an introduction to the basic principles of computer science (CS) from the perspective of mobile computing, including programming in App Inventor, a graphical programming language for Android mobile devices. The lessons and materials used by students incorporate programming while also integrating all other AP CSP big ideas: creativity, abstraction, data and information, algorithms, the internet and global impact. The curriculum engages students and supports the development of problem solving skills honing in on the computational thinking practices as indicated in the AP CSP curriculum framework. Students learn to create socially useful computational artifacts using App Inventor as well as connect computing and learn about abstracting as they develop and analyze their programs. The curriculum also emphasizes communication and collaboration in a project-based approach and classroom environment. This course involves a strong writing component. Students will maintain a portfolio of their work, which will include several performance tasks in the areas of programming and the impact of computing technology.

Recommended Materials:

- USB Thumb drive to store back-up copies of assignment and project files

Grading:

Scale: {A(100-90), B(89-80), C(79-70), D(69-60), F(59-0)}

Round-Up for Quarter Grade Only: {A(90-89.5), B(80-79.5), C(70-69.5), D(60-59.5)}

Incomplete quarter grades will not be given. Students are expected to follow the Late Assignment policy listed below.

Grades should be available online through Aries. Grades evaluated in the following categories and weights; **Projects** (30%), **Participation** (10%), and **Lab Assignments** (20%), **Exams and Quizzes** (40%). Rubrics will be provided in class and on Google Classroom

Academic Honesty:

Students must complete their own work for all assignments. If a student is found to have copied or turned in work that is not theirs a grade of Zero will be given.

Late Assignment Policies:

Any assignment not turned in during your class period on the due date will be considered late.

- Students must complete and turn in a Late Assignment Grade Request Form in order for that assignment to be graded.

- A grade of 0(zero) will be given for assignments not turned in or attempted.
- Partial credit may be given for Late Assignments turned in within a week of the due date.
- Full credit may be given for Late Assignments due to an **Excused Absence** within a week of the due date.

ATTENDANCE:

You are responsible for making up work missed due to excused absences. You must make arrangements with me immediately upon returning to class. I will not remind you to make up missed assignments.

Regular on time attendance and active positive participation are essential in order to understand and appreciate the topics covered in this class. It is also a critical life-long work skill. Irregular attendance and/or excessive tardiness will seriously affect your achievement in this class and consequently your grade.

TARDIES: {Over the duration of the class – 2 Quarters}

- 1 tardy: Verbal Warning
- 2 tardies: Conference with student and phone call home
- 3 tardies: Referral

School related tardies are not to be counted as tardies for this policy

Class Rules:

- No Food or Drinks
 - Bottled water is okay
- Get to class On-Time
 - In your seat logging in
- No Disrespectful Behavior
 - Do not be mean or insulting
 - Do not alter or break the equipment

Class Procedures

Beginning of class:

- Be in your seat and start logging in before the bell rings.
- Log in to Google Classroom and Read the daily agenda
- Complete the daily Journal response, turn-in assignments, review instructional materials

Instruction Time/Homework:

- Most days there will be a short lesson to introduce new topics and demonstrate how the assignments need to be completed.
- If we are working on a long-term project the teacher will review where you should be and check your progress

Lab Time:

- Stay in your seats and work on your assignments and projects. If you have a question raise your hand and the teacher will assist you as soon as possible.
- If the teacher is busy write your name on the white board and the teacher will see in the order as your name is listed.
- If you finish the activities listed for the day check with the teacher. There will always be an assignment or project coming up that you can work on (Check Google Classroom). Do not work on assignments from other classes.

Turning In Assignments and Projects:

- Most assignments and projects will be graded at your desk; others will be collected from Google Classroom

Leaving the Classroom:

- If you need to use the bathroom just let the teacher know before you leave.
- If you have a pass to leave early or you are playing a sport, it is your responsibility to let the teacher before you leave. Please do not expect the teacher to remind you that you have an appointment.

Lab Computer Policies:

Not following the policies listed will result in disciplinary action listed below

- Students are not allowed to unplug cables and move the computers and/or monitors
- Removing and/or rearranging the keys on any keyboard is not allowed
- Students will be held responsible for any equipment that they damage and the right to use the equipment may be revoked as a result
- All students must follow the district Acceptable Use policies

DISCIPLINARY ACTION:

A sequence of steps will be followed for any circumstance or situation that interrupts or interferes with the learning/teaching atmosphere in the classroom or any other learning/teaching environment.

1. Visual warning
2. Student moved from computer and given written assignment
3. Phone call home and/or referral

Curriculum Units and Projects:

The units that follow interweave the six AP CS Principles Computational Thinking Practices of Connecting Computing, Creating Computational Artifacts, Abstracting, Analyzing Problems and Artifacts, Communicating, and Collaborating with the seven CS Principles Big Ideas of Creativity, Abstraction, Data, Algorithms, Programming, Internet, and Global Impact.

- ❖ Unit 1 - Getting Started: Preview & Set up
- ❖ Unit 2 - Introduction to Mobile Apps & Pair Programming
- ❖ Unit 3 - Creating Graphics & Images Bit by Bit
- ❖ Create - Programming Performance Task #1 (Practice)
- ❖ Unit 4 - Exploring Computing: Animation, Simulation, & Modeling
- ❖ Exam #1
- ❖ Explore - Impact of Computing Innovations Performance Task #1 (Practice)
- ❖ Unit 5 - Algorithms & Procedural Abstraction
- ❖ Explore - Impact of Computing Innovations Performance Task #2
- ❖ Unit 6 - Using and Analyzing Data & Information
- ❖ Unit 7 - Communication Through The Internet
- ❖ Create - Programming Performance Task #2
- ❖ Exam #2

Assessments:

Portfolios

In this course students will document their work on their portfolios. That is, they will post answers to reading questions, write-ups of hands-on tutorials, written responses to assigned readings, and documentation of creative programming projects on their personal portfolio page. Each student will create a portfolio using Google sites (<https://www.google.com/sites/overview.html>). The portfolios will promote collaboration and sharing -- students can learn from each other -- and will constitute a full record of what the students have done in the course that they can refer back to during and after the course and share with their friends and family. Portfolios will be graded periodically throughout the duration of the course.

Reading and Homework Assignments

There will be regular reading and/or out-of-class homework assignments. These may include reading a chapter from the textbook and/or completing a tutorial or worksheet. Brief, clear, and concise written responses to the study questions must be posted on students' portfolios.

Labs

This course will be taught in a computer lab. Students will have access to computers and mobile devices and any other necessary hardware, both during the class and during free periods. Students can work in the lab during their free periods. Internet access will be available to students throughout the course. In each unit, there will be at least three

labs designed to practice and/or reinforce key concepts. Some are unplugged and others are completed in an online development environment. Most are completed in App Inventor.

Projects

There will be two (2) creative programming projects in which students will use lab time to work both individually and collaboratively (in pairs) to create a socially useful mobile app that they propose (pitch), design, and implement. One of these will be a practice for the College Board's Create Performance Task.

The second will be the official College Board Create Performance Task. Twelve (12) hours of class time will be provided for completion of the official Create Performance Task.

There will also be two (2) written research projects that students will work on individually. These research projects will focus on examining a computing innovation that has impacted society. One will be a practice for the Explore Performance Task. The second will be the College Board's Explore Performance Task. Eight (8) hours of class time will be provided for completion of the official College Board Explore Performance Task.

Oral and Video Presentations

There will be approximately three (3) oral and/or videotaped presentations of students' projects during the course.

Quizzes and Exams

There will be periodic quizzes, typically to wrap up the end of each unit, and a midterm exam given during the course. There will be a comprehensive final exam. Quizzes will be hand written and/or electronic and exams will be electronic.

Self-Check and Live Coding Exercises

All lessons in this course are accompanied by short, interactive, self-check exercises that consist of multiple choice and fill-in question as well as automatically graded, live-coding, programming exercises. These assessments are considered an essential part of the learning process. These are hosted online and may be done individually or with the class as a whole. Each question or exercise includes detailed feedback and students may repeat the question or exercise until it is correct.

AP CS Principles Exam

Students who complete this course will be prepared to take the AP CS Principles Exam.

Unit 1: Getting Started: Preview and Set up (Creativity, Algorithms, & Impact)

Unit 1 of the course provides a brief overview of the Mobile CSP curriculum, emphasizing its main theme: learning the principles of computer science while building socially useful mobile apps. The hands-on work focuses on setting up the student's environment, including their programming environment and online portfolios. Students are led through the process of creating a Gmail account, registering on the App Inventor site, and setting up their Google sites portfolio. Their portfolios will be used to display and share all of their written work for the course. Students are provided a brief introduction to blocks-based programming by having them work through a series of increasingly challenging Blockly Maze problems. And they are given a brief introduction to the Blown to Bits book, which is used as a reading resource throughout the course.

Unit 2: Introduction to Mobile Apps and Pair Programming (Creativity, Abstraction, Programming, & Impact)

Unit 2 provides an introduction to the App Inventor programming platform and the course's first programming project, the I Have a Dream app, a sound board app. Students are introduced to App Inventor's event-driven programming model. Students first work through a guided tutorial that plays an excerpt of a Martin Luther King speech and are then presented with several exercises that challenge them to extend their understanding by solving problems on their own, working in pairs. This is followed later in the unit by several creative mini projects where students are invited to express their own ideas by developing their own computational artifacts. Students are also introduced to several important CS Principles themes and topics. Two lessons focus on hardware and software concepts. The big idea of abstraction is introduced. Students get their first look at binary numbers learning how to count in binary and how to view number systems such as binary, hexadecimal and decimal, as instances of the higher-order abstraction of a positional number system.

Unit 3: Creating Graphics & Images Bit by Bit (Creativity, Abstraction, Data and Information, Programming, & Impact)

Unit 3 extends the student's mobile programming toolkit to several new App Inventor components and introduces a number of new programming concepts, including the concept of a variables, lists and data abstraction. The main app in this unit, The Paint Pot app, a computational version of finger painting, focuses on App Inventor's drawing

and painting features and related topics from the CS Principles framework. The app is presented in four parts each of which is followed by a set of creative project exercises and challenges. This unit also introduces two other apps: Magic 8 Ball app, which provides a first introduction to lists, and Map Tour, which demonstrates how to incorporate external data into a mobile app. Unit 3 also extends the student's understanding of binary number system and introduces students to the idea of a bit as the fundamental unit of data. Through a number of hands-on and interactive activities students explore how bits are used to represent images, and how redundant parity bits can be used to detect simple data transmission errors. These lessons are complemented nicely by a Blown to Bits reading that focuses on digital documents, including how information can be hidden inside images and other digital documents.

Create: Programming Performance Task #1 (Creativity, Abstraction, Algorithms, & Programming)

Up until this point students have completed App Inventor tutorials and they have been given smaller challenges. This programming task is a practice for the official Create programming performance task that will be submitted to the College Board. Students are given 12-15 hours of class time to complete this task.

Assessment: Create Your Own Mobile App

Students work collaboratively with a partner (pair programming) to create a socially useful, interactive, mobile app. The app must in some way include drawing, graphics, and programming constructs based on skills learned in prior lessons. Students are taught how to brainstorm their ideas and develop wire-frames with storyboards to express those ideas. Students are asked to give a 1-2 minute elevator pitch of their app idea and receive feedback from the instructor and their classmates. In class time is given to develop, test, and debug their app. The instructor answers any questions and provides feedback along the way. While working on their app, students are shown how to and asked to maintain a portfolio write up of their work making note of their progress and any challenges they may have faced, as well as, screenshots of blocks of code with written explanations of the how the code works. Students are shown how to record a video of their app. The project ends with an in class presentation and app demo by each pair of students.

Unit 4: Animation, Simulation, and Modeling: Exploring the Impact of Computing (Creativity, Abstraction, Data and Information, Algorithms, Programming, & Impact)

Unit 4 focuses on animation, simulation and modeling. The Android Mash app introduces the idea of computer simulation with a computational version of the traditional Whack-a-Mole game. The Coin Flip app, which extends over several lessons, introduces the concept of modeling. The activities in Unit 4 build toward EU 2.3 as students learn that models use abstractions, such as a pseudo random number generator (PRNG), to represent real world situations, in this case, the flipping of a coin; EU 3.3 as students learn how PRNG algorithms are used to model randomness inside a computer, such as with the Coin Flip app; EU 7.1 as students extend the app model to represent different types of coins, including a biased coin and a three-sided coin. This is followed by an experimental lesson where an app that repeatedly “flips” a coin is used to assess the quality of App Inventor’s PRNG; EU 7.3 as students learn how one’s privacy is impacted by developing technology and computing innovations; and EU 7.4 as students learn the economic, social and cultural effects of computing innovations, such as real world models of the weather and the solar system.

Explore: Impact of a Computing Innovation Performance Task #1 (Creativity, Impact)

Up until this point students have read Blown to Bits chapters and excerpts, as well as, read and discussed articles about recent computing innovations that have been in the news. Students are encouraged to find daily news articles about advances in technology and share them with the class. This task is a practice for the official Explore performance task that will be submitted to the College Board. Students are given 4-5 hours of class time to complete this activity.

Activity: Impact of a computing innovation

This activity involves discussing, as a class, a computing innovation that has had considerable impact on the social, economic, or cultural areas of our lives, such as phone monitoring software. Students work collaboratively in small groups to research the computing innovation and find reliable sources using sites such as the ACM Digital Library. Students are also asked to cite their sources and are instructed about plagiarism. The instructor assigns each group member a prompt taken from the official Explore Performance Task to answer about the innovation. Each group member answers the prompts in a single Google document that is shared among the group. The group then works together to edit the entire document discussing changes that need to be made. When the document is completed (i.e. all prompts are answered and all sources are cited), each student is asked prepare their own original digital artifact (e.g. music, image, video, infographic, presentation, program, web page) to express the effects the

chosen innovation. Students are asked to share their artifact with their group members. After completing this activity, the students are asked to reflect on the experience and to brainstorm at least three computing innovations they might want to research for the official Explore Performance Task

Unit 5: Algorithms and Procedural Abstraction (Abstraction, Algorithms, Programming, & Impact)

In Unit 5, algorithms and procedures are examined in more detail. The Logo apps introduce the concept of procedural abstraction and students learn to define and use procedures -- named blocks of code that perform a specific task. By encapsulating the algorithms into named procedures and introducing parameters to help generalize the algorithms, students are led to see the advantages of procedural abstraction. In addition to designing and testing their own algorithms, students are also provided an introduction into the analysis of algorithms. Algorithm efficiency is examined for searching and sorting algorithms, which are analyzed both experimentally and through mathematical concepts such as functions and graphs. The impact section of this unit focuses on the impact that Web searching algorithms have had on our lives. The activities completed in Unit 5 build toward EU 2.2, EU 4.1, EU 4.2, EU 5.3 and EU 5.5 by focusing on abstraction, algorithms, and programming concepts.

Unit 6: Using and Analyzing Data and Information (Creativity, Data and Information, Programming, & Impact)

Unit 6 focuses on various aspects of using and manipulating Data, both within mobile apps and on the Web and Internet. The App Inventor lessons in this unit focus on different types of programming data, including variables and structured data, such as lists and databases. Students build apps that involve persistent data, data that persists from one instance of the app to another, and learn how to share data online by using simple Application Programming Interfaces (APIs), such as the Google Fusion table API. This unit's CS Principles lessons build toward EU 3.1, EU 3.2, EU 7.1, EU 7.2, and EU 7.5 by focusing on the concept of Big Data and its growing importance and its impact on society. Students are also introduced to some of the algorithms for processing massive data sets.

Unit 7: Communication Through The Internet (Creativity, Programming, The Internet, & Impact)

Unit 7 focuses on the Internet, one of the big ideas in computer science. The App Inventor lessons in this unit show different ways to use the internet in apps, including

the ability to send text messages over Wifi, finding directions via the Google Maps API. The CS Principles lessons focus on the Internet, how it works, how it enables innovation and collaboration, and security concerns for using it.

Welcome to the class! I look forward to working with you