NEW MILFORD PUBLIC SCHOOLS

New Milford, Connecticut

AP Computer Science Principles

June 2018

BOE Approved September 2018

New Milford Board of Education

Superintendent of Schools

Mr. Joshua Smith

Assistant Superintendent

Ms. Alisha DiCorpo

Authors of Course Guide:

Shana Bergonzelli-Graham

BOE Approved September 2018

New Milford's Mission Statement

The mission of the New Milford Public Schools, a collaborative partnership of students, educators, family and community, is to prepare each and every student to compete and excel in an ever-changing world, embrace challenges with vigor, respect and appreciate the worth of every human being, and contribute to society by providing effective instruction and dynamic curriculum, offering a wide range of valuable experiences, and inspiring students to pursue their dreams and aspirations.

Curriculum Author's Forward

This course is supported by the Mobile Computer Science Principles Project (Mobile CSP), an NSF-funded effort to provide a broad and rigorous introduction to computer science based on App Inventor, a mobile programming language for Android devices. The course is based on the College Board's emerging Advanced Placement (AP) Computer Science Principles curriculum framework for introductory computer science.
In this course, students will learn computer science by building socially useful mobile apps. In addition to programming and computer science principles, the course is project-based and emphasizes writing, communication, collaboration, and creativity.

Major Units and Pacing Guides

Unit	Estimated Start Date	Estimated Timing
Unit 1	End of August	180 Minutes - done during summer and 1 80 min. class period during week 1.
Unit 2	Beginning of September	585 Minutes - Seven to eight 80 min class periods
Unit 3	Beginning of October	585 Minutes - Seven to eight 80 min class periods
Unit 4	Beginning of November	585 Minutes - Four to five 80 min class periods
Create #1 Performance Task	Mid December	360 Minutes - Seven to eight 80 min class periods
Exam 1 - Midterm*	Mid January	135 Minutes - 1 review period, 1 exam
Explore #1 Performance Task	Mid January	360 Minutes - Four to five 80 min class periods
Unit 5	End of January	540 Minutes - Seven 80 min class periods
Unit 6	Mid February	585 Minutes - Seven to eight 45 min class periods
Explore #2**	Beginning of March (uploaded to College Board by April 30th)	480 Minutes - Six to seven 80 min class periods
Unit 7	Mid March	495 Minutes - Six to seven 45 min class periods

Create #2	**Beginning of April (uploaded to College Board by April 30th)**	**720 Minutes - Nine 80 min class periods**
Unit 8	**Beginning of May**	**135 Minutes - Two 80 min class review sessions**
Final Exam	**Beginning of May**	**45 Minutes - One 80 min class period**
AP CSP Exam	**May 11th**	**Two hour long exam**
Unit 9	**Mid May**	**Extra App Inventor lessons as well as suggestions for other resources to engage students with future CS courses, majors, and careers**

Unit 1: Getting Started with MobileCSP: Preview and Setup

<table>
<tr><th colspan="3">Stage 1 Desired Results</th></tr>
<tr><td rowspan="4">ESTABLISHED GOALS

<u>APCSP Computational Thinking Practice 4: Analyzing Problems and Artifacts</u>
Evaluate a proposed solution to a problem.
Locate and correct errors
Explain how an artifact functions
Justify appropriateness and correctness of a solution, model, or artifact.
<u>APCSP Learning Objectives:</u>

<u>Big Idea 1 Creativity:</u>
Learning Objective 1.1.1 Apply a creative development process when creating computational artifacts

Learning Objective 1.2.1 Create a computational artifact for creative expression.

<u>Big Idea 4 Algorithms:</u>
Learning Objective 4.1.1 Develop an algorithm for implementation in a program

Learning Objective 4.1.2 Express an algorithm in a language.

Learning Objective 4.2.4 Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity.

<u>Big Idea 5 Programming:</u>
Learning Objective 5.1.2 Develop a correct</td><td colspan="2">Transfer</td></tr>
<tr><td colspan="2">Students will be able to independently use their learning to…

Follow instructions explicitly and precisely.

Understand that he brain is malleable and a growth mindset helps to achieve success.

Work collaboratively and independently to solve problems by applying the problem solving process.

State and support an opinion with clarity and specificity.</td></tr>
<tr><td colspan="2">Meaning</td></tr>
<tr><td>UNDERSTANDINGS
Students will understand that…

APCSP Big Idea 4: Algorithms
4.1 Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages.
4.2 Algorithms can solve many, but not all, computational problems.

APCSP Big Idea 5: Programming
5.1 Programs can be developed for creative expression, to satisfy personal curiosity, to create new knowledge, or to solve problems (to help people, organizations, or society).
5.2 People write programs to execute algorithms.</td><td>ESSENTIAL QUESTIONS
Students will keep considering…

Is coding a creative pursuit? How?

How are algorithms implemented and executed on computers and computational devices?

How are programs used for creative expression, to satisfy personal curiosity or to create new knowledge?

How do computer programs implement algorithms?

Are there problems that can’t be solved by algorithms?</td></tr>
</table>

<table>
<tr><td rowspan="2">program to solve problems.

Learning Objective 5.1.3 Collaborate to develop a program.

Learning Objective 5.2.1 Explain how programs implement algorithms

<u>APCSP Essential Knowledge</u>

<u>Big Idea 1 Creativity:</u>
EK 1.1.1A A creative process in the development of a computational artifact can include, but is not limited to, employing nontraditional, nonprescribed techniques; the use of novel combinations of artifacts, tools, and techniques; and the exploration of personal curiosities

EK 1.2.1A A computational artifact is something created by a human using a computer and can be, but is not limited to, a program, an image, audio, video, a presentation, or a Web page file..

<u>Big Idea 4 Algorithms:</u>
EK 4.1.1A Sequencing, selection, and iteration are building blocks of algorithms.

EK 4.1.2A Languages for algorithms include natural language, pseudocode, and visual and textual programming languages.

EK 4.2.4C The correctness of an algorithm is determined by reasoning formally or mathematically about the algorithm. (Exclusion statement: Formally proving program correctness is beyond the scope of this course and the AP Exam)</td><td colspan="2">Acquisition</td></tr>
<tr><td>Students will know…

● Essential vocabulary: Integrated Development Environment (IDE), algorithm, program, iteration and control structure
● All programs are made of control structures: sequence, selection and repetition
● A cloud application runs completely within a browser environment (code is not on the machine).
● AppInventor is an example of an IDE. It is also a cloud application.
● A user interface is the part of a computer program that the user interacts with.
● You can design a program’s user interface in AppInventor’s designer mode and code the functionality for the program in AppInventor’s Blocks Editor mode.</td><td>Students will be skilled at…

● Developing an algorithm to solve a problem, evaluating the algorithm for correctness, implementing the algorithm and reviewing/explaining their algorithm.
● Collaborating with others to solve problems using the pair programming model (navigator/driver)
● Creating a Google sites portfolio that they will use during the course to post their work (computational artifacts).
● Creating an app in the AppInventor environment and connecting to App Inventor with a device to run a test app.</td></tr>
</table>

Big Idea 5 Programming: EK 5.1.2A An iterative process of program development helps in developing a correct program to solve problems.). EK 5.1.3A Collaboration can decrease the size and complexity of tasks required of individual programmers. EK 5.2.1A Algorithms are implemented using program instructions that are processed during program execution. EK 5.2.1B Program instructions are executed sequentially. Common Core Reading Standards for Informational Texts (Grades 11-12): 1. Cite strong and thorough textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text, including determining where the text leaves matters uncertain. 2. Determine two or more central ideas of a text and analyze their development over the course of the text, including how they interact and build on one another to provide a complex analysis; provide an objective summary of the text. 3. Analyze a complex set of ideas or sequence of events and explain how specific individuals, ideas, or events interact and develop over the course of the text. 4. Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings; analyze how an author uses and refines the meaning of a key term or terms over the course of a text (e.g., how Madison defines faction in		

Federalist No. 10).		

Stage 2 – Evidence		
Code	**Evaluative Criteria**	**Assessment Evidence**
A, T, M	Teacher Rubric	PERFORMANCE TASK(S): GOAL: Students will create a test app that demonstrates successful use of the AppInventor environment and download to device. Students will collaborate to test the app. ROLE: Developer. Tester AUDIENCE: Users of the app SITUATION: Testing successful connection of the student device to the AppInventor environment PRODUCT: Small text to speech app STANDARD: **APCSP Computational Thinking Practice 4: Analyzing Problems and Artifacts** ● Evaluate a proposed solution to a problem. ● Locate and correct errors ● Explain how an artifact functions ● Justify appropriateness and correctness of a solution, model, or artifact.
A A, T M, T M, T	Teacher will check that student's name was successfully added to list Teacher checklist Teacher observation Teacher observation	OTHER EVIDENCE: Successful addition to Google Classroom site Successful creation of Google portfolio site using template Class reflection on pervasive themes in Blown to Bits text. Small group discussion of the seven "Big Ideas" in the course

Stage 3 – Learning Plan		
Code	***Pre-Assessment*** Informal conversation with student when giving out summer packet will determine if the student has acquired the skills necessary to successfully complete the necessary components of this unit without assistance over the summer.	
	Summary of Key Learning Events and Instruction	Progress Monitoring
A	Teacher will provide students with a packet that highlights the tasks below that need to be completed over the summer. This first unit on course setup and introduction of the course material and will be completed independently EXCEPT for the AppInventor setup.	Review the packet with students individually before the summer. Encourage students to sign up for Google Classroom before school is over.
A	Welcome to Mobile CSP Teacher will assign student to read the online course introduction. An overview of the Mobile CSP Course, explains the CS Principles project and why the course uses *mobile computing* for its programming (coding) component.	Observe that student read the overview during Day 1 discussion of course goals and objectives.
A, T	Google Account, Classroom and Portfolio Set Up. The course requires a Google email (e.g., gmail) account. A Google site is created and used to share student reflections and other course work. Students will join the class' google Classroom site.	Verify receipt of an email from each student over the summer with a link to the Google site.Verify Classroom signup. Use Remind app to remind students to complete task before school starts.
A, T	App Inventor Setup. App Inventor is a visual, blocks-based programming language for creating mobile Android apps. Teacher demonstrates using Google credentials to create an account on MIT App Inventor site and then use it to create a test app to test the mobile device.	Verify that a small test app has been created and uploaded to the student's portfolio to demonstrate successful setup of the AppInventor environment. Verify successful app download to mobile device.
A, M, T	BB: Blown to Bits. Takes a look at the free, online version of ***Blown to Bits***. Readings from this book will be used throughout the course to focus on important issues that highlight the impact of computing on society.	Observe that student skimmed the text during Day 1 discussion of course goals and objectives.
A, M, T	Direct students to watch Successful Learning in Mobile CSP. video.	Discussion questions to determine that student watched the video during Day 1 discussion of course goals and objectives.

Unit 2: Introduction to Mobile Apps and Pair Programming

<table>
<tr><th colspan="3">Stage 1 Desired Results</th></tr>
<tr><td rowspan="4">ESTABLISHED GOALS

Computational Thinking Practice 2: Creating Computational Artifacts
Create a computational artifact with a practical, personal, or societal intent.
Select appropriate techniques to develop a computational artifact.
Use appropriate algorithms and information management principles.

Computational Thinking Practice 3: Abstracting
Explain how data, information, or knowledge is represented for computational use.
Explain how abstractions are used in computation or modeling.
Identify abstractions
Describe modeling in a computational context.

Computational Thinking Practice 4: Analyzing Problems and Artifacts
Evaluate a proposed solution to a problem.
Locate and correct errors
Explain how an artifact functions
Justify appropriateness and correctness of a solution, model, or artifact.

Computational Thinking Practice 5: Communicating</td><td colspan="2">Transfer</td></tr>
<tr><td colspan="2">Students will be able to independently use their learning to…

Read a text and identify main idea and details.

Be able to successfully use a data abstraction.

Follow instructions explicitly and precisely.

Work collaboratively and independently to solve problems by applying the problem solving process.

State and support an opinion with clarity and specificity.</td></tr>
<tr><td colspan="2">Meaning</td></tr>
<tr><td>UNDERSTANDINGS
Students will understand that…

APCSP Big Idea 1: Creativity
1.2 Computing enables people to use creative development processes when using computing tools and techniques to create computational artifacts for creative expression of ideas or to solve a problem.

APCSP Big Idea 2: Abstraction
2.1 A variety of abstractions built upon binary sequences can be used to represent all digital data.
2.2 Multiple levels of abstraction are used to write programs or to create other computational artifacts.</td><td>ESSENTIAL QUESTIONS
Students will keep considering…

Is coding a creative pursuit? How?

How are algorithms implemented and executed on computers and computational devices?

Why are some languages better than others when used to implement algorithms?

What kinds of problems are easy, what kinds are difficult, and what kinds are impossible to solve algorithmically?

How are algorithms evaluated?

What is the Internet, how is it built, and how does it work?</td></tr>
</table>

Explain the meaning of a result in context. Describe computation with accurate and precise language, notations, or visualizations. Summarize the purpose of a computational artifact. **<u>APCSP Learning Objectives:</u>** **<u>Big Idea 1 Creativity:</u>** ● Learning Objective 1.1.1 Applying a creative development process when creating computational artifacts ● Learning Objective 1.2.1 Creating a computational artifact for creative expression. ● Learning Objective 1.2.3 Creating a new computational artifact by combining or modifying existing artifacts ● Learning Objective 1.2.4 Collaborating in the creation of computational artifacts. ● Learning Objective 1.3.1 Using computing tools and techniques for creative expression. **<u>Big Idea 2 Abstraction:</u>** ● Learning Objective 2.1.1 Describing the variety of abstractions used to represent data. ● Learning Objective 2.1.2 Explaining how binary sequences are used to represent digital data. ● Learning Objective 2.2.3 Identifying multiple levels of abstractions that are used when writing programs. **<u>Big Idea 4 Algorithms:</u>** ● Learning Objective 4.1.1 Developing an algorithm for implementation in a		What aspects of the Internet's design and development have enabled it to grow so large and be so influential? How does cybersecurity affect what we do on the Internet? How does abstraction make the development of computer programs possible? How do computer programs implement algorithms?

program ● Learning Objective 4.1.2 Expressing an algorithm in a language. **<u>Big Idea 5 Programming:</u>** ● Learning Objective 5.1.1 Developing a program for creative expression, to satisfy personal curiosity, or to create new knowledge ● Learning Objective 5.1.2 Developing a correct program to solve problems ● Learning Objective 5.2.1 Explaining how programs implement algorithms. ● Learning Objective 5.4.1 Evaluating the correctness of a program. **<u>Big Idea 6 The Internet:</u>** ● Learning Objective 6.1.1 Explaining the abstractions in the Internet and how the Internet functions. (Exclusion statement: Specific devices used to implement the abstractions in the Internet are beyond the scope of this course and the AP Exam.) **<u>Big Idea 7 Global Impact:</u>** ● Learning Objective 7.1.1 Explain how computing innovations affect communication, interaction, and cognition. ● Learning Objective 7.1.2 Explain how people participate in a problem-solving process that scales. ● Learning Objective 7.3.1 Analyze the beneficial and harmful effects of computing.	***APCSP Big Idea 4: Algorithms*** 4.1 Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages. ***APCSP Big Idea 5: Programming*** 5.1 Programs can be developed to solve problems (to help people, organizations, or society); for creative expression; to satisfy personal curiosity or to create new knowledge. 5.2 People write programs to execute algorithms. 5.4 Programs are developed, maintained, and used by people for different purposes. ***APCSP Big Idea 6: The Internet*** 6.1 The Internet is a network of autonomous systems ***APCSP Big Idea 7: Global Impact*** 7.1 Computing enhances communication, interaction, and cognition. 7.3 Computing has global effects — both beneficial and harmful — on people and society.	
	Acquisition	
	Students will know… ● **A Computational artifact** is an object created by a human being that involves the use of computation in some way, for example a mobile app or a web page. ● **Event-driven programming** is a programming approach whereby the program's behavior is controlled by	*Students will be skilled at…* ● Following an instructor-led walkthrough to create an app on a mobile device; ● Navigating the App Inventor programming platform ● Using event handlers in AppInventor ● Using the different drawers (folders) in AppInventor to code functionality for user events. ● Using a selection *if-else* block to pause

APCSP Essential Knowledge **Big Idea 1 Creativity:** ● EK 1.1.1A A creative process in the development of a computational artifact can include, but is not limited to, employing non-traditional, non-prescribed techniques; the use of novel combinations of artifacts, tools, and techniques; and the exploration of personal curiosities. ● EK 1.2.1A A computational artifact is something created by a human using a computer and can be, but is not limited to, a program, an image, audio, video, a presentation, or a Web page file. ● EK 1.2.1B Creating computational artifacts requires understanding of and use of software tools and services. ● EK 1.2.1C Computing tools and techniques are used to create computational artifacts and can include, but are not limited to, programming integrated development environments (IDEs), spreadsheets, 3D printers, or text editors. ● EK 1.2.1E Creative expressions in a computational artifact can reflect personal expressions of ideas or interests. ● EK 1.2.3A Creating computational artifacts can be done by combining and modifying existing artifacts or by creating new artifacts. ● EK 1.2.3C Combining or modifying existing artifacts can show personal expression of ideas. ● EK 1.2.4A A collaboratively created computational artifact reflects effort by more than one person. ● EK 1.3.1B Digital audio and music can be created by synthesizing sounds, sampling existing audio and music, and recording and manipulating sounds, including layering and looping.	writing code that responds to various events that occur, such as Button clicks. ● **Hardware** is the large and small physical components that make up a computers such as the computer's keyboard or its processor. ● **Software** is the computer programs that make up a computer system such as the mobile apps we will be creating in this course. ● **Abstraction** is one of the seven big ideas of the CS Principles curriculum. An abstraction is a simplified and general representation of some complex object or process. One example --we'll encounter many in this course, including abstractions used in computer programming -- would be a Google map. ● **Binary number** is a number written in the binary system, a system that uses only two digits, 0s and 1s. ● **a bit** is short for **bi**nary digi**t** ● **a byte** is 8 bits ● **blacklist** in internet terminology, it is a generic term for a list of email addresses or IP addresses that are origination with known spammer ● **a character** is any symbol that requires one byte of storage ● **data** is distinct information that is formatted in a special way. Data exists in a variety of forms, like text on paper or bytes stored in electronic memory ● **data center**s are physical or virtual infrastructures used by enterprises to house computer, server and networking systems and components for the company's IT (information technology) needs	and start the speeches. ● Reusing code efficiently by using the copy and paste blocks feature in App Inventor Blocks Editor ● Using a horizontal arrangement to layout buttons side-by-side ● Naming components in a standard format (description followed by component type, e.g. Malcolm Button) ● Giving an AppInventor app its own custom icon that will appear in the device's app launcher when the app is packaged (built) ● Using App Inventor's Text-to-Speech component (Media drawer) to get the AppInventor app to speak some words ● Having the app vibrate the device when an event is triggered ● Using App Inventor's Accelerometer Sensor (Sensor drawer)to trigger a Text-to-Speech when the device is shaken. ● Giving examples of abstractions in everyday life ● Converting between binary, hexadecimal, octal and decimal numbering systems ● Using the AND OR and NOR logic gates in a Logic.ly simulation to create circuits. ● Ordering related abstractions from most abstract to least abstract. ● Giving examples of everyday items that are stored as bits. ● Analyzing the positive or negative impacts of a computing innovation

Big Idea 2 Abstraction: ● EK 2.1.1A Digital data is represented by abstractions at different levels ● EK 2.1.1B At the lowest level, all digital data are represented by bits. ● EK 2.1.1C At a higher level, bits are grouped to represent abstractions, including but not limited to numbers, characters, and color. ● EK 2.1.1D Number bases, including binary, decimal, and hexadecimal, are used to represent and investigate digital data. ● EK 2.1.1F Hexadecimal (base 16) is used to represent digital data because hexadecimal representation uses fewer digits than binary. ● EK 2.1.1G Numbers can be converted from any base to another base. ● EK 2.1.2A A finite representation is used to model the infinite mathematical concept of a number. (Exclusion statement: Binary representations of scientific notation are beyond the scope of this course and the AP Exam). ● EK 2.1.2C In many programming languages, the fixed number of bits used to represent real numbers (as floating-point numbers) limits the range of floating-point values and mathematical operations; this limitation can result in round off and other errors. ● EK 2.1.2D The interpretation of a binary sequence depends on how it is used. ● EK 2.1.2E A sequence of bits may represent instructions or data. ● EK 2.1.2F A sequence of bits may represent different types of data in different contexts. ● EK 2.2.3A Different programming languages offer different levels of abstraction. (Exclusion statement: Knowledge of the abstraction capabilities of all programming languages is	● **a data network** is a telecommunications network which allows computers to exchange data ● **a disk drive** is a randomly addressable and rewritable storage device ● **intellectual property** refers to any property that is created using original thought. Traditional intellectual property include patents, copyrights, and trademarks. ● **Moore's Law**: refers to how the number of transistors per square inch on integrated circuits has doubled every year since the integrated circuit was invented. ● **a network** is a group of two or more computer systems linked together ● **a processor** is short for microprocessor or CPU ● **a social network** is a social structure made of nodes that are generally individuals or organizations. A social network represents relationships and flows between people, groups, organizations, animals, computers, or other information/knowledge processing entities ● **a whitelist** is a generic name for a list of email address or IP addresses that are considered to be spam free ● UI Components are parts of the user interface such as Buttons, Labels, etc ● The ***World Wide Web (WWW)*** is an application that runs on the Internet. The WWW is a system of interlinked resources -- documents, images, sounds, videos, data files -- that are stored on the Internet and can be accessed through a *browser*.	

beyond the scope of this course and the AP Exam). ● EK 2.2.3B High-level programming languages provide more abstractions for the programmer and make it easier for people to read and write a program. ● EK 2.2.3C Code in a programming languages is often translated into code in another (lower level) language to be executed on the computer. ● EK 2.2.3D In an abstraction hierarchy, higher levels of abstraction (the most general concepts) would be placed toward the top and lower level abstraction (the more specific concepts) toward the bottom. ● EK 2.2.3E Binary data is processed by physical layers of computing hardware, including gates, chips, and components. ● EK 2.2.3F A logic gate is a hardware abstraction that is modeled by a Boolean function. (Exclusion statement: Memorization of specific gate visual representations is beyond the scope of this course and the AP Exam). ● EK 2.2.3G A chip is an abstraction composed of low-level components and circuits that perform a specific function. ● EK 2.2.3H A hardware component can be low level like a transistor or high level like a video card. ● EK 2.2.3I Hardware is built using multiple levels of abstractions, such as transistors, logic gates, chips, memory, motherboards, special purpose cards, and storage devices. ● EK 2.2.3J Applications and systems are designed, developed, and analyzed using levels of hardware, software, and conceptual abstractions. ● EK 2.2.3K Lower level abstractions can be	● The ***Internet*** is the underlying global network that supports the WWW and many other applications. It consists of many different local networks that are connected together by various hardware devices. ● The ***Cloud*** is just a popular term for the Internet and its applications used largely in marketing and advertising. Facebook, Google, Twitter, Dropbox are often referred to as ***cloud applications***. They could also be called Internet applications. App Inventor is another example of a cloud application. ● ***Browsers***: are programs that display web pages and are used to navigate the WWW. ● A protocol is a system of rules that govern the behavior of some system. ● TCP/IP (Transmission Control Protocol/Internet Protocol) is the suite a protocols that determine the behavior of the Internet. ● HTTP (HyperText Transfer Protocol) is the protocol that controls the behavior of the WWW. ● Tim Berners-Lee invented the World Wide Web (WWW). ● Berners-Lee felt the WWW brought the Internet to a higher level of abstraction. ● An open standard is a standard (such as TCP, HTTP) that is not owned or controlled by a private entity. It stands in contrast to 'proprietary' materials', which are owned or controlled by a private entity. Open Standards fuel the growth of the Internet! ● The International Engineering Task Force (IETF) develops and oversees open	

combined to make higher level abstractions, such as short message services (SMS) or email messages, images, audio files, and videos. **<u>Big Idea 4 Algorithms:</u>** ● EK 4.1.1A Sequencing, selection, and iteration are building blocks of algorithms. ● EK 4.1.1B Sequencing is the application of each step in an algorithm in the order in which the statements are given. ● EK 4.1.1C Selection uses a Boolean condition to determine which of two parts of an algorithm is used. ● EK 4.1.1D Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times. ● EK 4.1.2A Languages for algorithms include natural language, pseudocode, and visual and textual programming languages. ● EK 4.1.2B Natural language and pseudocode describe algorithms so that humans can understand them. ● EK 4.1.2C Algorithms described in programming languages can be executed on a computer. ● EK 4.1.2D Different languages are better suited for expressing different algorithms. ● EK 4.1.2E Some programming languages are designed for specific domains and are better for expressing algorithms in those domains. ● EK 4.1.2F The language used to express an algorithm can affect characteristics such as clarity or readability but not whether an algorithmic solution exists. ● EK 4.1.2G Every algorithm can be constructed using only sequencing, selection, and iteration. ● EK 4.1.2H Nearly all programming languages	standards such as HTTP (www) and SMTP (mail). ● The Ten Commandments of Computer Ethics ● The if block in AppInventor can be used to choose between different actions ● 'Chip' is an informal way of describing an integrated circuit (IC) consisting of millions of tiny circuits. ● compilation is the process of translating the entire source code into a single binary file. ● A computer is a machine that processes information under the control of a program. ● CPU (Central Processing Unit) is that part of the computer's hardware that carries out the instructions of a computer program. ● A general purpose computer is one that can run many different programs (e.g. a smartphone). ● A computer's hardware includes its electronic and mechanical components that carries out the instructions of a computer program. ● A high level language is a programming language that is human readable (App Inventor) and provides the programmer with easy to understand abstractions. ● The process of translating source code into machine language one instruction at a time and immediately executing instruction. ● A machine language is a programming language that is directly readable by the computer’s CPU ● The motherboard houses the computer's main electronic	

are equivalent in terms of being able to express any algorithm. **Big Idea 5 Programming:** ● EK 5.1.1B Programs developed for creative expression, to satisfy personal curiosity, or to create new knowledge may have visual, audible, or tactile inputs and outputs. ● EK 5.1.2A An iterative process of program development helps in developing a correct program to solve problems. ● EK 5.1.2B Developing correct program components and then combining them helps in creating correct programs. ● EK 5.2.1F Processes use memory, a central processing unit (CPU), and input and output. ● EK 5.4.1M The functionality of a program is often described by how a user interacts with it. ● EK 5.5.1E Logical concepts and Boolean algebra are fundamental to programming. ● EK 5.5.1F Compound expressions using and, or, and not are part of most programming languages. **Big Idea 6 the Internet:** ● EK 6.1.1A The Internet connects devices and networks all over the world. ● EK 6.1.1D The Internet and the systems built on it facilitate collaboration. ● EK 6.1.1I Standards such as hypertext transfer protocol (HTTP), IP, and simple mail transfer protocol (SMTP) are developed and overseen by the Internet Engineering Task Force (IETF). **Big Idea 7 Global Impact:**	components. ● A computer program is a sequence of instructions that controls the computer. ● RAM (Random Access Memory) stores the computer's programs and data temporarily while power is on. ● Software consists of the programs that control the computer ● A special purpose computer is one that has a fixed program (e.g. a calculator, a watch, a car's brakes). ● A Boolean condition is a true/false condition. It is named after George Boole (1815-1864) an English mathematician. ● Pseudocode is a notation for expressing algorithms, which is more precise that ordinary English but less formal than a programming language. ● A flowchart is a visual (i.e. graphical) notation for expressing algorithms. ● App Inventor apps have a **5 Mb size limit**. ● Many sounds and images online are copyrighted and it is a violation of copyright to include such images in your app. So, you should be careful about the images and sounds you put into your apps. If you want to use a copyrighted image or sound in your app, you will have to get permission from the holder of the copyright. ● Abstracting is the process of creating abstractions ● A constant, such as the numeral '5', is an abstraction that represents a single thing, e.g., the value 5. ● A variable, such as the symbol 'X', can be used to represent any number and is	

• EK 7.1.1B Video conferencing and video chat have fostered new ways to communicate and collaborate. • EK 7.1.1D Cloud computing fosters new ways to communicate and collaborate. • EK 7.1.1O The internet and the Web have impacted productivity, positively and negatively, in many areas. • EK 7.1.2F Crowdsourcing offers new models for collaboration, such as connecting people with jobs and businesses with funding. • EK 7.3.1A Innovations enabled by computing raise legal and ethical concerns. • EK 7.3.1H Aggregation of information, such as geolocation, cookies, and browsing history, raises privacy and security concerns. • EK 7.3.1J Technology enables the collection, use, and exploitation of information about, by, and for individuals, groups, and institutions.	therefore more general and more abstract than a constant • Procedural abstraction in computer science is the practice of organizing and encapsulating algorithms in named procedures that can then be invoked by name. An example would be the 'sqrt(x)', square root of x, which encapsulates the algorithm for calculating the square root of x. • Data abstraction in computer science is the practice of organizing and encapsulating certain data into a more general representation. An example would be storing the text 'hello' in a single variable rather than having numerous occurrences of 'hello' in a program • The base of a number system refers to the number of distinct digits or symbols used to represent numbers in that system. Our decimal system is base-10 because it uses 10 digits, 0 through 9 • In a positional number system, such as our decimal system, the value of a digit in a number depends on its place. For example, in the decimal number 545, the leftmost '5' represents 500 because it occurs in the hundreds place, but the rightmost '5' represents 5 because it occurs in the ones place. • The octal numbering system represents all number using 8 digits (0-7) • The hexadecimal numbering system represents all numbers using 16 digits (0-9 and letters A-F) • Hexadecimal numbers are often preceded by 0x • Logic gates are the fundamental building	

	blocks of electronic circuits. • In an AND gate the output is TRUE (the light is ON) when both of its inputs are TRUE (or ON). • In an OR gate the output is TRUE (the light is ON) when either or both of its inputs are TRUE (or ON). • In a NOT gate the output is is TRUE (or ON) when its single input is FALSE (or OFF). • A transistor is a semiconductor device used to amplify or switch electronic signals and electrical power. Transistors are the fundamental building blocks of electronic devices. • A flip flop (or latch) is a digital circuit that has two states, ON or OFF, that can be used to store a 1 or a 0. It is the fundamental unit of computer memory	

Stage 2 – Evidence		
Code	**Evaluative Criteria**	**Assessment Evidence**
M	M/C, Completion and Short Ans.	Units 1+ 2 Quiz: Students should score an 80% on this 45 question quiz PERFORMANCE TASK(S):
T, M	Teacher Checklist and Rubric	I Have a Dream, Projects. Student-create a project in AppInventor, building on the "I Have a Dream" app previously completed in a tutorial. This app plays Martin Luther King's "I Have a Dream" speech among other inspiring speeches when a button with an an image is clicked. This student-customized project adds Text-to-speech and Accelerometer components to the app. Solution requires the use of an if/else algorithm (selection).
		OTHER EVIDENCE:
A	Teacher Checklist and Rubric	I Have a Dream, I. Student follows a tutorial to create an app that plays an MLK speech when a Button is clicked. Introduces event-driven programming.
A, M	Teacher Checklist and Rubric	I Have a Dream, II. Student follows a tutorial that extends the previous app to include two buttons, the second of which plays a Malcolm X speech. Introduces the if/else statement to toggle between playing and pausing the speech when a button is clicked.
M	Coding questions (correct/incorrect with feedback)	Unit 2 Basic App Inventor Drills Practice
M	Teacher Rubric	POGIL Activities: the Internet and the Cloud, Algorithm Basics and Hardware: Logic.ly App
A, T	Teacher Observation	Student Discussions: Blown to Bits Ch. 1

Stage 3 – Learning Plan		
Code	***Pre-Assessment*** Teacher created pre-assessment that assesses prior understanding of internet vocab, basic algorithm concepts, computer terminology, number system conversions and low level hardware concepts.	
	Summary of Key Learning Events and Instruction	Progress Monitoring
A	I Have a Dream, I. Teacher directs student to follow a tutorial to create an app that plays an MLK speech when a Button is clicked. Introduces event-driven programming.	Program Checklist for necessary elements
A, T, M	The Internet and the Cloud. Introduces some basic concepts about the Internet and the concept of cloud computing. Students read and discuss the Wikipedia article on *10 Commandments of Computer Ethics.* Videos and slides are available to present the material. Includes a POGIL activity to discuss browsers, online applications that students use and the difference between the Internet and WWW.	Student self-check. Student reflection. Teacher observation during discussion.
A, M	I Have a Dream, II. Teacher directs student to follow a tutorial that extends the previous app to include two buttons, the second of which plays a Malcolm X speech. Introduces the if/else statement to toggle between playing and pausing the speech when a button is clicked.	Program Checklist for necessary elements
A, M	Mobile Devices and Apps: Hardware and Software. Teacher plays video and shows slides that introduce computer terminology, such as hardware, software, operating system, programming languages.	Self Check Questions, Student Portfolio Reflection
A, M	Algorithm Basics. Teacher plays video and shows slides that cover basic algorithm concepts, including sequence, selection (if/else), repetition (loops). A POGIL activity asks students to write a simple arithmetic algorithm in pseudocode.	Self check questions. Student portfolio reflection. Teacher observation during discussion.
A, M	What is Abstraction? Teacher shows a video lecture that provides a first look at the concept of abstraction with examples drawn from everyday experience. Provides examples of data abstraction and procedural abstraction.	Self-check questions. Student portfolio reflection.
A, M	Binary Numbers. Teacher demonstrates a first look at the binary number system, focusing on how to count, how to convert binary to decimal and decimal to binary.	Monitor during interactive activity for correct/incorrect answers and provide feedback. Practice AP Question. Student self-check. Student

    A, M A, M, T	This lesson includes a number of videos that demonstrate conversions. It also introduces *hexadecimal (base 16)* numbers and the general concept of a *positional number system.* Includes several interactive widgets for practicing with binary and hex. Hardware Abstractions: Logic Gates. Teacher facilitates a second look at abstraction, this time focusing on low level hardware such as logic gates. A POGIL activity uses the **Logicly** app to study logic gates BB: The Digital Explosion. Teacher directs students to read Chapter 1 of ***Blown to Bits***, and leads a discussion which makes the point that today "everything is digital." Our music, our images and videos, our books are software are all represented as binary data.	reflection. Self-check questions. Student portfolio reflection. Teacher observation during discussion. Student portfolio reflection. Teacher observation during discussion (Debate Team Carousel format).

Unit 3: Creating Graphics and Images Bit by Bit

Stage 1 Desired Results

ESTABLISHED GOALS

Computational Thinking Practice 1: Connecting Computing

Identify impacts of computing.

Describe connections between people and computing.

Explain connections between computing concepts.

Computational Thinking Practice 2: Creating Computational Artifacts

Create a computational artifact with a practical, personal, or societal intent.
Select appropriate techniques to develop a computational artifact.
Use appropriate algorithms and information management principles.

Computational Thinking Practice 3: Abstracting

Explain how data, information, or knowledge is represented for computational use.
Explain how abstractions are used in computation or modeling.
Identify abstractions
Describe modeling in a computational context.

Computational Thinking Practice 4: Analyzing Problems and Artifacts

Evaluate a proposed solution to a problem.

Transfer

Students will be able to independently use their learning to…

Follow directions explicitly and specifically.

Seek constructive feedback and specific assistance from other classmates.

Work collaboratively and independently to solve problems by applying the problem solving process

Meaning

UNDERSTANDINGS
Students will understand that…

APCSP Big Idea 1: Creativity
1.2 Computing enables people to use creative development processes when using computing tools and techniques to create computational artifacts for creative expression of ideas or to solve a problem.

APCSP Big Idea 2: Abstraction
2.1 A variety of abstractions built upon binary sequences can be used to represent all digital data.
2.2 Multiple levels of abstraction are used to write programs or to create other computational artifacts.
2.3 Models and simulations use abstraction to generate new understanding and knowledge.

APCSP Big Idea 3: Data And Information
3.1 People use computer programs to process information to gain insight and knowledge.
3.2 Computing facilitates exploration and the discovery of connections in information.

ESSENTIAL QUESTIONS
Students will keep considering…

How do people develop and test computer programs?

Which mathematical and logical concepts are fundamental to computer programming?

How can computation be employed to help people process data and information to gain insight and knowledge?

How can computation be employed to facilitate exploration and discovery when working with data?

What considerations and trade-offs arise in the computational manipulation of data?

How are vastly different kinds of data, physical phenomena, and mathematical concepts represented on a computer?

Locate and correct errors Explain how an artifact functions Justify appropriateness and correctness of a solution, model, or artifact. **APCSP Learning Objectives:** **Big Idea 1 Creativity:** Learning Objective 1.2.2 Create a computational artifact using computing tools and techniques to solve a problem Learning Objective 1.2.4 Collaborate in the creation of computational artifacts. Learning Objective 1.3.1 Use computing tools and techniques for creative expression. **Big Idea 2 Abstraction:** Learning Objective 2.1.1 Describe the variety of abstractions used to represent data. Learning Objective 2.1.2 Explain how binary sequences are used to represent digital data. Learning Objective 2.2.1 Develop an abstraction when writing a program or creating other computational artifacts. Learning Objective 2.2.2 Develop an abstraction when writing a program or creating other computational artifacts. Learning Objective 2.3.1 Use models and simulations to represent phenomena **Big Idea 3 Data and Information:**	3.3 There are trade-offs when representing information as digital data. ***APCSP Big Idea 5: Programming*** 5.1 Programs can be developed to solve problems (to help people, organizations, or society); for creative expression; to satisfy personal curiosity or to create new knowledge. 5.2 People write programs to execute algorithms. 5.3 Programming is facilitated by appropriate abstractions. 5.4 Programs are developed, maintained, and used by people for different purposes. 5.5 Programming uses mathematical and logical concepts. ***APCSP Big Idea 7: Global Impact*** 7.1 Computing enhances communication, interaction, and cognition. 7.3 Computing has global effects — both beneficial and harmful — on people and society.	
	Acquisition	
	Students will know… ● When components are added to the AppInventor viewer, they are laid out vertically by default. ● Variables are named containers that stand for values that can change ● Increment means to add one to a variable ● all digital data, including electronic documents, are composed of bits, ● representing an image digitally is another example of abstraction at work.	*Students will be skilled at…* ● Coding buttons to change the paint colors ● Using App Inventor’s touch and drag event handlers to draw circles and lines on the canvas. ● Using a variable to make a program more general. ● Adding components to the AppInventor Viewer. ● Naming variables appropriately ● Setting properties of a component

Learning Objective 3.1.2 Collaborate when processing information to gain insight and knowledge Learning Objective 3.1.3 Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language. Learning Objective 3.2.1 Extract information from data to discover and explain connections, patterns, or trends. Learning Objective 3.3.1 Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. **Big Idea 4: Algorithms** Learning Objective 4.1.2 Express an algorithm in a language **Big Idea 5 Programming:** Learning Objective 5.1.1 Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge Learning Objective 5.1.2 Develop a correct program to solve problems Learning Objective 5.1.3 Collaborate to develop a program Learning Objective 5.2.1 Explain how programs implement algorithms.	● ASCII is short for American Standard Code for Information Interchange is a character encoding scheme in which each character is represented by a 7-bit (originally) or 8-bit binary sequence. For example, the ASCII sequence 01000001 represents the letter 'A ● A bitmap is a type of memory organization or image file format used to store digital images ● A lossless compression algorithm is one in which no data are lost; the original data can be completely recovered ● A lossy compression algorithm is one in which some data are lost; the original data cannot be completely restored ● A compression algorithm that represents an image in terms of the length of runs of identical pixels ● RLE is a compression algorithm that represents an image in terms of the length of runs of identical pixels ● The process of refactoring means to revise the code, leaving the app’s behavior unchanged. ● A good practice to follow is to provide comments in the following situations: to document every procedure that you define and to clarify a complex algorithm that isn’t clearly obvious. ● Representing data binary sequences can be used to detect certain errors in data. ● adding extra bits to a binary sequence can be used for error detection. ● Redundant parity bits can be used to detect representation and transmission errors in data. ● An API stands for Application Programming Interface and their role is to specify exactly how programs and apps can interact with each other to perform certain tasks, like sending	● Changing a component to fill its container ● Understanding ASP pseudocode for variables ● Using a setter to change the value of variables after they have been initialized ● Incrementing a variable using a setter ● Using a getter to retrieve the value of a variable ● Explaining how images are compressed by using RLE (Run-Length-Encoding) ● Describing the difference between lossless and lossy compression ● Adding a custom image to an AppInventor app by uploading it into the designer. ● Adding a button to support a new color option for an AppInventor app. ● Creating a button that decrements a variable by using a setter ● Using an if/else algorithm to perform input validation to ensure a variable does not go below or above its specified range (fixes an existing bug in the program). ● Creating a button that resets the value of a variable by storing the original value in a second variable. ● Using the camera component to upload an image to the app. ● Defining a programmer-defined procedure that will help reduce the complexity of code and make it easier to read and maintain. ● Documenting the code by using the add comment option so it is easier for other people to understand and read ● Describing what “code smell” is ● Explaining how errors are detected when data is transmitted using technology like a parity bit. ● Explaining how positive and negative parity works. ● Using the Google Maps API through the ActivityStarter component to perform different map-related functions in an app

Learning Objective 5.3.1 Use abstraction to manage complexity in programs. Learning Objective 5.4.1 Evaluate the correctness of a program. Learning Objective 5.5.1 Employ appropriate mathematical and logical concepts in programming. **Big Idea 7 Global Impact:** Learning Objective 7.1.1 Explain how computing innovations affect communication, interaction, and cognition. Learning Objective 7.3.1 Analyze the beneficial and harmful effects of computing. **APCSP Essential Knowledge:** **Big Idea 1 Creativity:** EK 1.2.2A Computing tools and techniques can enhance the process of finding a solution to a problem. EK 1.2.4A A collaboratively created computational artifact reflects effort by more than one person. EK 1.2.4C Effective collaborative teams practice interpersonal communication, consensus building, conflict resolution, and negotiation. EK 1.2.4D Effective collaboration strategies enhance performance. EK 1.2.4E Collaboration facilitates the application of multiple perspectives (including sociocultural perspectives) and diverse talents and skills in	email or Twitter messages or displaying a map. The API specifies exactly what information you need to provide and in what specific format to provide it in order to interact with an existing application. ● Google Maps API provides documentation for programmers and app developers on how to interact with its application ● Cryptography means secret writing. It is the science of protecting information by transforming it into an unreadable format ● An analog device or system is one that represents changing values as continuously variable physical quantities ● A digital system is any system based on discontinuous data or events. Computers are digital machines because at the basic level they can distinguish between just two values, 0 and 1. ● To download data is to copy data (usually an entire file) from an online source to a personal computer. ● A megabyte (MB) is a unit for characterizing the amount of data. It is roughly 1 million bytes or, more precisely, 220 bytes, which is 1,048,576 bytes ● A megapixel is one million pixels, used in reference to the resolution of a graphics device. ● Modeling is the process of representing a real-world object of phenomenon as a set of mathematical equations ● Optical character recognition (OCR) is the process of reading text from paper and translating the images into a form that the computer can manipulate. ● A raster is the rectangular area of a display screen actually being used to display images. ● A pixel is short for a picture element, a single	● Creating an app that uses a List to store and access data ● Creating an app that randomly selects items from a list. ● Creating an app that uses App Inventor's GPS sensor to determine the user's location and use it as a destination; ● Creating an app that uses a TinyDB component to save info in a list. ● Explaining the advantages of storing data persistently using TinyDB instead of in memory using variables. ● Explaining the difference between a raster image and an ASCII representation of a text document ● Providing an example of free and open source software ● Providing examples of real-life models ● Giving an example of how a piece of metadata could increase the usefulness of an image or document.

developing computational artifacts. EK 1.2.4F A collaboratively created computational artifact can reflect personal expressions of ideas. EK 1.3.1C Digital images can be created by generating pixel patterns, manipulating existing digital images, or combining images. **<u>Big Idea 2 Abstraction:</u>** EK 2.1.1B At the lowest level, all digital data are represented by bits. EK 2.1.1C At a higher level, bits are grouped to represent abstractions, including but not limited to numbers, characters, and color. EK 2.1.1E At one of the lowest levels of abstraction, digital data is represented in binary (base 2) using only combinations of the digits zero and one. (Exclusion statement: Two's complement conversions are beyond the scope of this course and the AP Exam). EK 2.1.2B In many programming languages, the fixed number of bits used to represent characters or integers limits the range of integer values and mathematical operations; this limitation can result in overflow or other errors. (Exclusion statement: Range limitations of any one language, compiler, or architecture are beyond the scope of this course and the AP Exam). EK 2.1.2D The interpretation of a binary sequence depends on how it is used. EK 2.1.2F A sequence of bits may represent different types of data in different contexts.	point in a graphic image. ● To render refers to the process of adding realism to a computer graphics by adding 3-D qualities, such as shadows and variations in color and shade. ● Spam is electronic junk mail or junk newsgroup postings. ● Steganography is the art and science of hiding information by embedding messages within other, seemingly harmless messages. ● To upload data means to transmit data from a computer to an online repository or service such as a bulletin board service, dropbox, or network	

EK 2.2.1A The process of developing an abstraction involves removing detail and generalizing functionality. EK 2.2.1B An abstraction extracts common features from specific examples in order to generalize concepts. EK 2.2.2B Being aware of and using multiple levels of abstraction in developing programs helps to more effectively apply available resources and tools to solve problems. EK 2.3.1A Models and simulations are simplified representations of more complex objects or phenomena. EK 2.3.1B Models may use different abstractions or levels of abstraction depending on the objects or phenomena being posed. **Big Idea 3 Data and Information:** EK 3.1.2B Collaboration facilitates solving computational problems by applying multiple perspectives, experiences, and skill sets. EK 3.1.2E Collaborating face-to-face and using online collaborative tools can facilitate processing information to gain insight and knowledge. EK 3.1.3E Interactivity with data is an aspect of communicating. EK 3.2.1G Metadata is data about data. EK 3.2.1H Metadata can be descriptive data about an image, a Web page, or other complex objects.		

EK 3.2.1I Metadata can increase the effective use of data or data sets by providing additional information about various aspects of that data. EK 3.3.1C There are trade-offs in using lossy and lossless compression techniques for storing and transmitting data. EK 3.3.1D Lossless data compression reduces the number of bits stored or transmitted but allows complete reconstruction of the original data. EK 3.3.1E Lossy data compression can significantly reduce the number of bits stored or transmitted at the cost of being able to reconstruct only an approximation of the original data. EK 3.3.1G Data is stored in many formats depending on its characteristics (e.g., size and intended use). EK 3.3.1I Reading data and updating data have different storage requirements. **Big Idea 4 Algorithms:** EK 4.1.2I Clarity and readability are important considerations when expressing an algorithm in a language. **Big Idea 5: Programming:** EK 5.1.1B Programs developed for creative expression, to satisfy personal curiosity, or to create new knowledge may have visual, audible, or tactile inputs and outputs.		

EK 5.1.2A An iterative process of program development helps in developing a correct program to solve problems. EK 5.1.2B Developing correct program components and then combining them helps in creating correct programs. EK 5.1.2D Program documentation helps programmers develop and maintain correct programs to efficiently solve problems. EK 5.1.2E Documentation about program components, such as blocks and procedures, helps in developing and maintaining programs. EK 5.1.2F Documentation helps in developing and maintaining programs when working individually or in collaborative programming environments. EK 5.1.3A Collaboration can decrease the size and complexity of tasks required of individual programmers. EK 5.1.3B Collaboration facilitates multiple perspectives in developing ideas for solving problems by programming. EK 5.1.3C Collaboration in the iterative development of a program requires different skills than developing a program alone. EK 5.1.3D Collaboration can make it easier to find and correct errors when developing programs. EK 5.1.3E Collaboration facilitates developing program components independently. EK 5.1.3F Effective communication between participants is required for successful		

collaboration when developing programs. EK 5.2.1C Program instructions may involve variables that are initialized and updated, read, and written. EK 5.3.1A Procedures are reusable programming abstractions. EK 5.3.1B A procedure is a named grouping of programming instructions. EK 5.3.1C Procedures reduce the complexity of writing and maintaining programs. EK 5.3.1D Procedures have names and may have parameters and return values. EK 5.3.1I Strings and string operations, including concatenation and some form of substring, are common in many programs EK 5.3.1J Integers and floating-point numbers are used in programs without requiring understanding of how they are implemented. EK 5.4.1A Program style can affect the determination of program correctness. EK 5.4.1B Duplicated code can make it harder to reason about a program. EK 5.4.1C Meaningful names for variables and procedures help people better understand programs. EK 5.4.1D Longer code blocks are hard to reason about than shorter code blocks in a program.		

EK 5.4.1E Locating and correcting errors in a program is called debugging the program. EK 5.4.1G Examples of intended behavior on specific inputs help people understand what a program is supposed to do. EK 5.5.1A Numbers and numerical concepts are fundamental to programming. EK 5.5.1D Mathematical expressions using arithmetic operators are part of most programming languages. **Big Idea 7: Global Impact** EK 7.1.1I Global Positioning System (GPS) and related technologies have changed how humans travel, navigate, and find information related to geolocation. EK 7.3.1A Innovations enabled by computing raise legal and ethical concerns. EK 7.3.1H Aggregation of information, such as geolocation, cookies, and browsing history, raises privacy and security concerns. EK 7.3.1J Technology enables the collection, use, and exploitation of information about, by, and for individuals, groups, and institutions.		

Stage 2 – Evidence		
Code	**Evaluative Criteria**	**Assessment Evidence**
M M A, M, T	M/C, Completion and Short answer Teacher created checklist and rubric Teacher created checklist and rubric	Unit 3 Quiz: Students should score at least an 80% Performance Tasks: Paint Pot Projects. Students modify the existing Paint Pot App to add basic refinements plus the use of Camera component to add a real-time image as the Canvas background. Programming refinements to add decrement button plus using and if/else algorithm to prevent the radius from becoming negative. Paint Pot Refactoring and Documentation. Introduces the concepts of refactoring and procedural abstraction. A student-created procedure is used in an app to encapsulate an algorithm that is used 3 times in the app, thereby reducing complexity. Also illustrates how to add comments to App Inventor blocks. Students keep the functionality of the app but learn that it is important to make the app easier for other programmers to understand and use.
 M M A, T	 Correct completion of drills (can retry) Teacher Observation Teacher Observation	OTHER EVIDENCE: AppInventor Drills: ● Setters, Getters, and Math -- drills that use App Inventor's *variable* and *setter* and *getter* blocks together with simple math operations from the *Math* drawer. ● If/Else Drills -- coding drills that focus on *if/else* statements and logic. ● Code Refactoring -- exercises that focus on revising and existing to incorporate *procedural abstraction* by defining *procedures* and *functions*. POGIL Activities: Error Detection Blown to Bits Chapter 3 discussion

Stage 3 – Learning Plan		
Code	***Pre-Assessment*** Short Answer focusing on program documentation, data storage (bits and bytes), image representation and parity.	
	Summary of Key Learning Events and Instruction	Progress Monitoring
A, T	Paint Pot Tutorial. Teacher directs students to independently complete App Inventor's version of the classic finger painting app. Introduces Canvas touched and dragged events. Introduces global variables for storing and incrementing the radius of the dots drawn. Teacher circulates to provide assistance and enforces "ask three before me" rule.	App Completion checklist.
A	Representing Images. Teacher uses a *CS Unplugged* lesson to show how bits (0s and 1s) are used to represent images. Students practice applying the *image compression* technique known as *run-length encoding (RLE).* Also, provides a brief introduction to ASCII.	Monitoring during unplugged activity and practice. Teacher observation during discussion.Student self-check and reflection
A, T	Error Detection. Teacher uses a *CS Unplugged* lesson (the card trick) to introduce the concept of using redundant bits in data to help detect errors. A POGIL activity asks students to work in teams to figure out how the trick works.	Monitoring during unplugged activity and practice. Teacher observation during discussion.Student self-check and reflection
M, T	Parity Error Detection. A follow-up lesson to that introduces the concept of *parity error checking*, with exercises on even- and odd-parity. Students read independently then self-reflect and practice.	Practice exercises. Student self-check and reflection
A, T	Map Tour Tutorial Teacher directs students to independently complete the tutorial. Uses Activity Starter component to display a location on a Google map. Teacher circulates to provide assistance and enforces "ask three before me" rule.	App Completion checklist.
M, T	Map Tour with GPS and Tiny DB tutorial. Teacher directs students to independently complete this tutorial which uses the GPS tool and incorporates a database	App Completion checklist.

	component so information can be stored. Teacher circulates to provide assistance and enforces “ask three before me” rule.	
A, M, T	BB: Electronic Documents. Students read and discuss part of Chapter 3 of ***Blown to Bits***, which focuses on *modeling* as it applies to *image representation.* Also introduces the concept of *steganography* -- i.e., hiding information in documents. An image editor widget is used to let the student hide their initials in a Bitmap, giving practice with binary sequences and ASCII codes.	Teacher observation during discussion. Check off completed bitmap with initials. Student Reflection.

Unit 4: Animation, Simulation and Modeling

<table>
<tr><th colspan="3">Stage 1 Desired Results</th></tr>
<tr><td rowspan="3">ESTABLISHED GOALS

Computational Thinking Practice 1: Connecting Computing
Identify impacts of computing.
Describe connections between people and computing.
Explain connections between computing concepts.

Computational Thinking Practice 2: Creating Computational Artifacts
Create a computational artifact with a practical, personal, or societal intent.
Select appropriate techniques to develop a computational artifact.
Use appropriate algorithms and information management principles.

Computational Thinking Practice 3: Abstracting
Explain how data, information, or knowledge is represented for computational use.
Explain how abstractions are used in computation or modeling.
Identify abstractions
Describe modeling in a computational context.

Computational Thinking Practice 4: Analyzing Problems and Artifacts
Evaluate a proposed solution to a problem.</td><td colspan="2">Transfer
Students will be able to independently use their learning to…

Analyze pros and cons of options in a real life situation

Test hypotheses

Gather data and support a hypothesis as valid or invalid

Work collaboratively and independently to solve problems by applying the problem solving process.</td></tr>
<tr><td colspan="2">Meaning</td></tr>
<tr><td>UNDERSTANDINGS
Students will understand that…

APCSP Big Idea 1: Creativity
1.2 Computing enables people to use creative development processes when using computing tools and techniques to create computational artifacts for creative expression of ideas or to solve a problem.
1.3 Computing can extend traditional forms of human expression and experience.

APCSP Big Idea 2: Abstraction
2.2 Multiple levels of abstraction are used to write programs or to create other computational artifacts.
2.3 Models and simulations use abstraction to generate new understanding and knowledge.

APCSP Big Idea 3: Data And Information
3.3 There are trade-offs when representing information as digital data.</td><td>ESSENTIAL QUESTIONS
Students will keep considering…

What are some potential beneficial and harmful effects of computing?

How do economic, social, and cultural contexts influence innovation and the use of computing?
How are vastly different kinds of data, physical phenomena, and mathematical concepts represented on a computer?

How does abstraction help us in writing programs, creating computational artifacts and solving problems?

How can computational models and simulations help generate new understanding and knowledge?

How do people develop and test computer programs?</td></tr>
</table>

<table>
<tr>
<td rowspan="3">Locate and correct errors
Explain how an artifact functions
Justify appropriateness and correctness of a solution, model, or artifact.

Computational Thinking Practice 5: Communicating
Explain the meaning of a result in context.
Describe computation with accurate and precise language, notations, or visualizations.
Summarize the purpose of a computational artifact.

Computational Thinking Practice 6: Collaborating
Collaborate with another student in solving a computational problem.
Collaborate with another student in producing an artifact.
Share the workload by providing individual contributions to an overall collaborative effort.
Foster a constructive, collaborative climate by resolving conflicts and facilitating the contributions of a partner or team member.
Exchange knowledge and feedback with a partner or team member.
Review and revise their work as needed to create a high-quality artifact.

<u>APCSP Learning Objectives:</u>

<u>Big Idea 1 Creativity:</u>

Learning Objective 1.2.2 Create a computational artifact using computing tools and techniques to solve a problem</td>
<td>APCSP Big Idea 4: Algorithms

4.1 Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages.

APCSP Big Idea 5: Programming
5.1 Programs can be developed to solve problems (to help people, organizations, or society); for creative expression; to satisfy personal curiosity or to create new knowledge.
5.3 Programming is facilitated by appropriate abstractions.
5.5 Programming uses mathematical and logical concepts.

APCSP Big Idea 7: Global Impact
7.1 Computing enhances communication, interaction, and cognition.
7.3 Computing has global effects — both beneficial and harmful — on people and society.
7.4 Computing innovations influence and are influenced by the economic, social, and cultural contexts in which they are designed and used</td>
<td>Which mathematical and logical concepts are fundamental to computer programming?</td>
</tr>
<tr>
<td colspan="2">Acquisition</td>
</tr>
<tr>
<td>Students will know…

● ad hoc, when used to describe programming, means a quick fix for a problem, not usually the best example that will sustain an issue.
● cloud computing is comparable to grid computing, and relies on sharing resources rather than having local servers handle applications.</td>
<td>Students will be skilled at…

● Using the Canvas and ImageSprites in App Inventor
● Creating an app with a timer event component
● Creating an app that incorporates randomness
● Creating an app with a socially useful theme</td>
</tr>
</table>

Learning Objective 1.2.4 Collaborate in the creation of computational artifacts. Learning Objective 1.3.1 Use computing tools and techniques for creative expression. **Big Idea 2 Abstraction:** Learning Objective 2.2.1 Develop an abstraction when writing a program or creating other computational artifacts. Learning Objective 2.2.3 Identify multiple levels of abstractions that are used when writing programs. Learning Objective 2.3.1 Use models and simulations to represent phenomena Learning Objective 2.3.2 Use models and simulations to formulate, refine, and test hypotheses **Big Idea 3 Data and Information:** Learning Objective 3.3.1 Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. **Big Idea 4: Algorithms** Learning Objective 4.1.1 Develop an algorithm for implementation in a program Learning Objective 4.1.2 Express an algorithm in a language **Big Idea 5 Programming:**	● a cookie is a small text file placed when you access a site and used by websites to track your activity on their site. A cookie allows the website to store and easily look up your records in their archive. ● a database is a collection of information organized in such a way that a computer program can quickly selected the desired pieces of data. Often abbreviated DB ● data aggregation is a process in which information is gathered and expressed in a summary form for purposes such as statistical analysis ● data mining is a class of database applications that look for hidden patterns in a group of data that could be used to predict future behavior ● a data repository generically refers to a general place where data is stored and maintained ● data sources is name given to the connection setup from a database to a server. The name is commonly used when creating a query to the database ● digital detritus is term used to describe unsightly debris that accrues as the result of the experience of digital living ● a dossier is a collection of documents about a person, event, or subject ● EDR stands for event data recorder ● to encode means to prepare data for storage or transmission. ● encryption is the translation of data into secret code ● geotagging is the process of adding geographical information to various media in the form of metadata. The data usually consists of coordinates like latitude and longitude, but may even include bearing,	● Using an if/else selection algorithm on the timer-Enabled property of a Clock Component that starts/stops an action ● Using procedural abstraction to define a procedure that encapsulates reusable functionality (reset Game) ● Using primitive Logo commands to create algorithms to draw simple shapes ● Defining simple procedures to simplify the drawing process. ● Using loops to replace repeated commands. ● Expressing an algorithm in pseudocode ● Creating an artifact that uses randomness and simulates a model ● Creating a simple model of a coin flip ● Using random number blocks to generate a random value in a specific range ● Defining a global variable and assigning it an initial value; ● Using a conditional statement, If/Else, to evaluate a variable and follow an algorithm based on the value of a variable ● Using a For each number loop to repeatedly simulate the flipping of the coin and count the number of heads that occur. ● Using software to conduct an experiment; ● Making and testing a hypothesis about App Inventor's ability to generate random numbers. ● Using a Google spreadsheet to record and evaluate data ● Practicing modular arithmetic ● Predicting pseudo-random numbers in a sequence ● Explaining the fetch-execute cycle. ● Explaining how a computer processes machine language instructions. ● Creating simple programs in machine language using a simulator

Learning Objective 5.1.1 Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge Learning Objective 5.1.2 Develop a correct program to solve problems Learning Objective 5.1.3 Collaborate to develop a program Learning Objective 5.2.1 Explain how programs implement algorithms. Learning Objective 5.3.1 Use abstraction to manage complexity in programs. Learning Objective 5.5.1 Employ appropriate mathematical and logical concepts in programming. **<u>Big Idea 7 Global Impact:</u>** Learning Objective 7.1.1 Explain how computing innovations affect communication, interaction, and cognition. Learning Objective 7.3.1 Analyze the beneficial and harmful effects of computing. Learning Objective 7.4.1 Explain the connections between computing and economic, social and cultural context. **<u>APCSP Essential Knowledge:</u>** **<u>Big Idea 1 Creativity:</u>** EK 1.2.2A Computing tools and techniques can enhance the process of finding a solution to a problem.	altitude, distance and place names. ● IP address is an identifier for devices on a TCP/IP network ● ISP stands for Internet Service Provider ● metadata is data about data; describes how and when and by whom a particular set of data was collected, and how data is formatted ● PRISM is a secret program or tool that performs data collection for the NSA ● a query is request for information from a database ● RFID is radio frequency identification, similar to barcodes ● a server is a computer program or a device that provides functionality for other programs or devices, called "clients". A server can be used to share data or resources among multiple clients or to perform computation ● a model is an abstraction that provides a simplified representation of some complex object or phenomenon. ● Randomness is the lack of pattern or regularity. A random sequence of events has no order or pattern ● a random event is an event that cannot be predicted with certainty. Examples would include flipping a fair coin, rolling a die, picking a card from a well shuffled deck ● A random number generator (PRNG) is an algorithm that generates a sequence of numbers that seem to occur in random order. ● PRNGs are models of true randomness. As such, they can be 'good' or 'bad' depending on how well they approximate true randomness. Much research by mathematicians and computer scientists goes into creating good PRNGs.	● Summarizing the differences between assembly language and machine language programming. ● Analyzing the differences between digital documents and paper documents ● Describing the responsibilities app developers have for the data collected. ● Analyzing the effectiveness of the Privacy Act.

EK 1.2.4A A collaboratively created computational artifact reflects effort by more than one person. 1.2.4C Effective collaborative teams practice interpersonal communication, consensus building, conflict resolution, and negotiation. EK 1.2.4D Effective collaboration strategies enhance performance. EK 1.2.4E Collaboration facilitates the application of multiple perspectives (including sociocultural perspectives) and diverse talents and skills in developing computational artifacts. EK 1.2.4F A collaboratively created computational artifact can reflect personal expressions of ideas. EK 1.3.1A Creating digital effects, images, audio, video, and animations has transformed industries. EK 1.3.1E Computing enables creative exploration of both real and virtual phenomena. EK 1.3.1D Digital effects and animations can be created by using existing software or modified software that includes functionality to implement the effects and animations. **<u>Big Idea 2 Abstraction:</u>** EK 2.2.1A The process of developing an abstraction involves removing detail and generalizing functionality. EK 2.2.1B An abstraction extracts common features from specific examples in order to generalize concepts.	● App Inventor uses a standard and well established PRNG, which should do a good job of modeling randomness. ● A hypothesis is an explanation that can be tested by experimentation ● The ENIAC is the first digital computer. ● ALU (Arithmetic Logic Unit) is that part of the CPU that performs all logic and arithmetic operations. ● The fetch-execute cycle is the basic process performed by the CPU. On each cycle the CPU fetches the next instruction from RAM, interprets it and executes it ● The instruction register is a special memory location in the CPU that stores the current instruction that is being executed. ● The instruction counter is a special register in the CPU that keeps track of the next instruction to be fetched. ● The accumulator is a special register in the CPU where data is put in order to perform arithmetic and logic operations ● An assembly language is low-level language that uses symbolic names, rather than binary sequences of 0s and 1s, to represent the machine language instructions	

EK 2.2.3A Different programming languages offer different levels of abstraction. (Exclusion statement: Knowledge of the abstraction capabilities of all programming languages is beyond the scope of this course and the AP Exam). EK 2.2.3C Code in a programming languages is often translated into code in another (lower level) language to be executed on the computer. EK 2.2.3J Applications and systems are designed, developed, and analyzed using levels of hardware, software, and conceptual abstractions. EK 2.3.1A Models and simulations are simplified representations of more complex objects or phenomena. EK 2.3.1B Models may use different abstractions or levels of abstraction depending on the objects or phenomena being posed. EK 2.3.1C Models often omit unnecessary features of the objects or phenomena that are being modeled. EK 2.3.1D Simulations mimic real-world events without the cost or danger of building and testing the phenomena in the real world. EK 2.3.2A Models and simulations facilitate the formulation and refinement of hypotheses related to the objects or phenomena under consideration. EK 2.3.2B Hypotheses are formulated to explain the objects or phenomena being modeled. EK 2.3.2C Hypotheses are refined by examining the insights that models and simulations provide		

into the objects or phenomena. EK 2.3.2D The results of simulations may generate new knowledge and new hypotheses related to the phenomena being modeled. EK 2.3.2E Simulations can facilitate extensive and rapid testing of models. EK 2.3.2F Simulations can facilitate extensive and rapid testing of models. EK 2.3.2G The time required for simulations is impacted by the level of detail and quality of the models and the software and hardware used for the simulation. EK 2.3.2H Rapid and extensive testing allows models to be changed to accurately reflect the objects or phenomena being modeled. **<u>Big Idea 3 Data and Information:</u>** EK 3.3.1A Digital data representations involve trade-offs related to storage, security, and privacy concerns. EK 3.3.1F Security and privacy concerns arise with data containing personal information. **<u>Big Idea 4 Algorithms:</u>** EK 4.1.1B Sequencing is the application of each step in an algorithm in the order in which the statements are given. EK 4.1.1C Selection uses a Boolean condition to determine which of two parts of an algorithm is used.		

EK 4.1.1D Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times. EK 4.1.1E Algorithms can be combined to make new algorithms. EK 4.1.1F Using existing correct algorithms as building blocks for constructing a new algorithm helps ensure the new algorithm is correct. EK 4.1.1G Knowledge of standard algorithms can help in constructing new algorithms. EK 4.1.2A Languages for algorithms include natural language, pseudocode, and visual and textual programming languages. EK 4.1.2C Algorithms described in programming languages can be executed on a computer. **Big Idea 5 Programming:** EK 5.1.1B Programs developed for creative expression, to satisfy personal curiosity, or to create new knowledge may have visual, audible, or tactile inputs and outputs. EK 5.1.2C Incrementally adding tested program segments to correct working programs helps create large correct programs. EK 5.1.3A Collaboration can decrease the size and complexity of tasks required of individual programmers. EK 5.1.3B Collaboration facilitates multiple perspectives in developing ideas for solving		

problems by programming. EK 5.1.3C Collaboration in the iterative development of a program requires different skills than developing a program alone. EK 5.1.3D Collaboration can make it easier to find and correct errors when developing programs. EK 5.1.3E Collaboration facilitates developing program components independently. EK 5.2.1D An understanding of instruction processing and program execution is useful for programming. EK 5.2.1E Program execution automates processes. EK 5.1.3F Effective communication between participants is required for successful collaboration when developing programs. EK 5.3.1A Procedures are reusable programming abstractions. EK 5.3.1B A procedure is a named grouping of programming instructions. EK 5.3.1C Procedures reduce the complexity of writing and maintaining programs. EK 5.5.1A Numbers and numerical concepts are fundamental to programming. EK 5.5.1B Integers may be constrained in the maximum and minimum values that can be represented in a program. (Exclusion statement: Specific range limitations of all programming		

languages are beyond the scope of this course and the AP Exam). EK 5.5.1C Real numbers are approximated by floating-point representations that do not necessarily have infinite precision. (Exclusion statement: Specific sets of values that cannot be exactly represented by floating point numbers are beyond the scope of this course and the AP Exam). **<u>Big Idea 7 Global Impact:</u>** EK 7.1.1M The Internet and the Web have enhanced methods of and opportunities for communication and collaboration. EK 7.1.1N The Internet and the Web have changed many areas, including e-commerce, health care, access to information and entertainment, and online learning. EK 7.3.1G Privacy and security concerns arise in the development and use of computational systems and artifacts. EK 7.3.1H Aggregation of information, such as geolocation, cookies, and browsing history, raises privacy and security concerns. EK 7.3.1I Anonymity in online interactions can be enabled through the use of online anonymity software and proxy servers. EK 7.3.1K People can have instant access to vast amounts of information online; accessing this information can enable the collection of both individual and aggregate data that can be used and collected. EK 7.3.1L Commercial and governmental curation		

of information may be exploited if privacy and other protections are ignored. EK 7.4.1A The innovation and impact of social media and online access varies in different countries and in different socioeconomic groups. EK 7.4.1B Mobile, wireless, and networked computing have an impact on innovation throughout the world.		

Stage 2 – Evidence		
Code	**Evaluative Criteria**	**Assessment Evidence**
M	M/C, Completion and Short answer	Unit 4 Quiz
M, T	M/C, Completion and Short answer	**Mobile CSP Exam 1** is the midterm exam for the course. This exam covers Units 1-4. PERFORMANCE TASK(S):
M, T	College Board Scoring Guidelines	**CREATE PT 1** is a practice programming performance task to prepare for the final one submitted to the College Board. The CREATE task is one of two required performance tasks by the College Board - a programming one (CREATE) and a written one (EXPLORE).In this programming performance task, students work in pairs to collaboratively develop a mobile app. This includes going through the entire development process of designing, implementing, and debugging a mobile app. Students then document their work by creating a portfolio write-up and share their work through an oral presentation to the class or a recorded video presentation.
M, T	College Board Scoring Guidelines	EXPLORE PT 1 is a practice EXPLORE impact of a computing innovation performance task to prepare for the final one submitted to the College Board. The EXPLORE task is one of two required through-course assessments by the College Board - a programming one (CREATE) and a written one (EXPLORE). In this practice written performance task, students work independently to research a computing innovation related to mobile apps that has had significant impact (both positive and negative) on our society. This includes finding credible, reliable, and recent sources, as well as answering a series of prompts about their chosen innovation. Students then create a visual artifact that demonstrates what they learned about one or more of the effects of the innovation. Note that this task should be considered scaffolding for the official task. Students will work in small groups
		OTHER EVIDENCE:
M	Correct Completion of Drills (Students can retake)	AppInventor Drills ● ***if-else statements*** ● ***math computations.***
M, T	Teacher Observation and Rubric	POGIL Activities: Coin flip and Real World Models
A, M, T	Teacher Observation	Blown to Bits Chapter 2 discussion

Stage 3 – Learning Plan		
Code **M, A, T**	***Pre-Assessment*** Students work in small groups to brainstorm examples of models and simulations in real life and uses of random numbers in real-life applications	
	Summary of Key Learning Events and Instruction	Progress Monitoring
A	LightsOff Tutorial Teacher directs students to independently complete this variation of the classic whack-a-mole game. Introduces animation and first use of a procedure definition. A Clock.Timer event is used to move the sprite to random locations on the Canvas. Teacher circulates to provide assistance and enforces "ask three before me" rule.	App completion Checklist
A, T, M	Students work in groups of two in the navigator-driver roles to complete the LightsOff Projects. Projects include adding a score keeping feature and increasing the sprite's speed as the score increases. Students to complete without scaffolding.	App completion Checklist
A, M	Logo 1. Teacher directs students to independently complete this app based on a template. The template provides a Logo-like drawing platform restricted to very primitive forward() and right turn() commands -- i.e., commands without parameters. (Parameters are introduced in Part II.) Problems include various sized squares and a face. The commands are too weak to draw a triangle, a shortcoming remedied in Part II. Introduces a counting loop to simplify expression of drawing algorithms. Teacher circulates to provide assistance and enforces "ask three before me" rule.	App completion Checklist
A, M	Coin Flip Simulation Tutorial. Teacher directs students to independently complete this modeling app to simulate a coin flip. Uses a global variable to represent the coin, App Inventor's random integer function to generate a 1 or 2, and an if/else algorithm to display heads or tails. Teacher circulates to provide assistance and enforces "ask three before me" rule.	App completion Checklist
A, M	Coin Flip Experiment. This is a lesson about ***modeling.*** In lesson 4.4 students wrote the Coin Flip app, which simulates flipping a coin. Teacher facilitates an experiment	Successful completion of experiment. Teacher observation during pair discussions and POGIL. Student Reflection

	to test the hypothesis that *App Inventor's random number generator is a* ***good model of random behavior***. POGIL activity is used to conduct the experiment, where an app is used to simulate 100s of coin flips. Students tabulate results and reflect on whether they support the hypothesis.	
A, M , T	Pseudo Random Numbers. Teacher uses slides and a video to explain how computers use an algorithm to generate number sequences that *seem* random. Introduces the concept of *modular arithmetic* or *clock arithmetic.* Involves some math (long division, modular arithmetic, evaluating an equation. Teacher models solving math equations.	Student self-check questions. Student Reflection in portfolio.
A, M, T	Students work in groups of two in the navigator-driver roles to complete the Coin Flip Simulation Projects. The projects extend the modeling begun in the coin flip tutorials. New app features (shaking) as well as new models -- 3-sided coin, biased coin. Students to complete without scaffolding.	App completion Checklist
A, M	Real World Models. Teacher uses online lectures and a video to demonstrate examples of modeling and simulation using real world examples such as climate models, models of the solar system, casino slot machines. Incorporates a POGIL activity to explore an interactive predator/prey model.	Teacher observation during POGIL. Student Reflection
M	Abstraction: Inside the CPU. Teacher directs students to a web app to simulate a 4-bit computer, with 16 bytes of RAM, a CPU with Accumulator and other registers. Illustrates progress through higher levels of abstraction as different generations of the simulation focus on the fetch-execute cycle, machine language programming, assembly language programming.	Student self-check questions. Student Reflection in portfolio.
A, M, T	BB: Privacy. Students read and teacher leads discussion of Chapter 2 of *Blown to Bits*, which focuses on the issue of how our privacy is affected by the digital explosion.	Teacher observation during discussion. Student Reflection in website portfolio.

Unit 5: Algorithms and Procedural Abstraction

<table>
<tr><th colspan="3">Stage 1 Desired Results</th></tr>
<tr><td rowspan="3">ESTABLISHED GOALS

Computational Thinking Practice 1: Connecting Computing
Identify impacts of computing.
Describe connections between people and computing.
Explain connections between computing concepts.

Computational Thinking Practice 2: Creating Computational Artifacts
Create a computational artifact with a practical, personal, or societal intent.
Select appropriate techniques to develop a computational artifact.
Use appropriate algorithms and information management principles.

Computational Thinking Practice 4: Analyzing Problems and Artifacts
Evaluate a proposed solution to a problem.
Locate and correct errors
Explain how an artifact functions
Justify appropriateness and correctness of a solution, model, or artifact.

<u>APCSP Learning Objectives:</u></td><td colspan="2">Transfer
Students will be able to independently use their learning to…

Work collaboratively and independently to solve problems by applying the problem solving process.

Analyze the effectiveness and efficiency of real-life algorithms using data tables and graphs as evidence.</td></tr>
<tr><td colspan="2">Meaning</td></tr>
<tr><td>UNDERSTANDINGS
Students will understand that…

APCSP Big Idea 2: Abstraction.
2.2 Multiple levels of abstraction are used to write programs or to create other computational artifacts.

APCSP Big Idea 4: Algorithms

4.1 Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages.
4.2 Algorithms can solve many but not all problems.

APCSP Big Idea 5: Programming
5.1 Programs can be developed to solve problems (to help people, organizations, or society); for creative expression; to satisfy personal curiosity or to create new knowledge.</td><td>ESSENTIAL QUESTIONS
Students will keep considering…

How are algorithms evaluated?

What kinds of algorithms are easy, what kinds are difficult and what kinds are impossible to solve algorithmically?

Which mathematical and logical concepts are fundamental to computer programming?</td></tr>
</table>

<table>
<tr>
<td rowspan="3"><u>Big Idea 2 Abstraction:</u>

Learning Objective 2.2.1 Develop an abstraction when writing a program or creating other computational artifacts.

Learning Objective 2.2.2 Use multiple levels of abstraction to write programs
Learning Objective 2.2.3 Identify multiple levels of abstractions that are used when writing programs.

<u>Big Idea 3 Data and Information:</u>

Learning Objective 3.2.1 Extract information from data to discover and explain connections, patterns, or trends.

<u>Big Idea 4: Algorithms</u>

Learning Objective 4.1.1 Develop an algorithm for implementation in a program

Learning Objective 4.1.2 Express an algorithm in a language

Learning Objective 4.2.1 Explain the difference between algorithms that run in a reasonable time and those that do not run in a reasonable time. (Exclusion statement: Any discussion of nondeterministic polynomial (NP) is beyond the scope of this course and the AP Exam).

Learning Objective 4.2.2 Explain the difference between solvable and unsolvable problems in computer science. (Exclusion statement: Determining whether a given problem is solvable or unsolvable is beyond the scope of this course</td>
<td>5.3 Programming is facilitated by appropriate abstractions.
5.4 Programs are developed, maintained, and used by people for different purposes.

APCSP Big Idea 7: Global Impact
7.1 Computing enhances communication, interaction, and cognition.</td>
<td></td>
</tr>
<tr>
<td colspan="2">Acquisition</td>
</tr>
<tr>
<td>Students will know…
a background: multitasking computers are capable of executing several tasks, or programs, at the same time
binary: pertaining to a number system that has just two unique digits
bot: short for robot, a computer program that runs automatically.
cache: a special high-speed storage mechanism
firewall: a part of a computer system or network that is designed to prevent unauthorized access to or from that network
foreground: in multiprocessing systems, the process that is currently accepting input from the keyboard or other input device
HTML: hypertext markup language, a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages
URL: (uniform resource locator) it is the global address of documents and other resources on the World Wide Web
Bubble and merge sort are comparison</td>
<td>Students will be skilled at…
Using more complex Logo commands to draw shapes;
Incorporating parameters into procedures
Defining their own procedures, their own abstractions, to draw more complex shapes.
Defining a pseudocode algorithms that will play a guessing game.
Explaining how the sequential search algorithm works
Explaining how the binary search algorithm works.
Explaining how the bubble sort algorithm works
Explaining how the merge sort algorithm works
Explaining how the bucket sort algorithm works
Explaining how the radix sort algorithm works.
Analyzing the various sort algorithms for efficiency but using data tables and graphs as evidence.
Comparing the various sort algorithms for</td>
</tr>
</table>

and AP Exam.) Learning Objective 4.2.3 Explain the existence of undecidable problems in computer science. Learning Objective 4.2.4 Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. **<u>Big Idea 5 Programming:</u>** Learning Objective 5.1.2 Develop a correct program to solve problems Learning Objective 5.2.1 Explain how programs implement algorithms. Learning Objective 5.3.1 Use abstraction to manage complexity in programs. Learning Objective 5.4.1 Evaluate the correctness of a program. Learning Objective 5.5.1 Employ appropriate mathematical and logical concepts in programming. **<u>Big Idea 7 Global Impact:</u>** Learning Objective 7.1.1 Explain how computing innovations affect communication, interaction, and cognition. Learning Objective 7.1.2 Explain how people participate in a problem-solving process that scales Learning Objective 7.3.1 Analyze the beneficial and harmful effects of computing. Learning Objective 7.5.1 Access,manage, and	sorts. Radix and bucket sorts are not. ● There are two senses in which an algorithm cannot solve a problem: ○ Undecidable Problems. There are certain problems which are the theoretically impossible to solve — by any algorithm. ○ Intractable Problems. There are problems that are practically impossible to solve in a reasonable time — i.e., there are known algorithmic solutions, but the algorithms are too inefficient/slow to solve the problem when the number of inputs grows large. ● A heuristic algorithm is one that provides a solution to a problem, although in many cases the solution may not be the best possible solution -- i.e., it may not be an optimal solution. ● brute force is solve by trial and error; trying every possible option ● The Halting Problem is the theoretical problem of determining whether a computer program will halt (produce an answer) or loop forever on a given input ● The traveling salesman problem is: given a list of cities and the distances between them find the shortest path visiting each city once and returning to the start. ● A reasonable time to solve a problem can be expressed in terms of a polynomial, given the inputs ● An unreasonable time to solve a problem can be expressed in terms of an exponent	number of swaps and passes. ● Describe the characteristics of a comparison sort. ● Creating an app that navigates through a list using an index variable, selects items from a list and checks for the end of the list; ● Working with parallel lists. ● Counting right/wrong answers using a list to keep track of which questions in an app have already been answered ● Using loops with lists ● Personalizing and customizing a given app. ● Creating a plot graph that compares the results of running various searching and sorting algorithms. ● Collaborating with others to create a valid heuristic to solve the traveling salesman problem ● Using myactivity.google.com to identify individual user trends when browsing and searching the internet. ● Using trends.google.com to identify aggregate trends when browsing and searching the internet. ● Identifying the positive and negative impacts of using trends to make predictions ● Explaining how "The architecture of human knowledge has changed as a result of search."

attribute information using effective strategies. **<u>APCSP Essential Knowledge:</u>** **<u>Big Idea 1 Creativity:</u>** EK 1.2.1A A computational artifact is something created by a human using a computer and can be, but is not limited to, a program, an image, audio, video, a presentation, or a Web page file. EK 1.2.2A Computing tools and techniques can enhance the process of finding a solution to a problem. **<u>Big Idea 2 Abstraction:</u>** EK 2.2.1C An abstraction generalizes functionality with input parameters that allow software reuse.,(Exclusion statement: An understanding of the difference between value and reference parameters is beyond the scope of this course and the AP Exam. EK 2.2.2A Software is developed using multiple levels of abstractions, such as constants, expressions, statements, procedures, and libraries. EK 2.2.3A Different programming languages offer different levels of abstraction. (Exclusion statement: Knowledge of the abstraction capabilities of all programming languages is beyond the scope of this course and the AP Exam). EK 2.2.3B High-level programming languages provide more abstractions for the programmer and make it easier for people to read and write a program. **<u>Big Idea 3 Data and Information:</u>**		

EK 3.1.1A Computers are used in an iterative and interactive way when processing digital information to gain insight and knowledge. EK 3.2.1D Search tools are essential for efficiently finding information. **<u>Big Idea 4 Algorithms:</u>** EK 4.1.1A High-level programming languages provide more abstractions for the programmer and make it easier for people to read and write a program. EK 4.1.1D Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times. EK 4.1.1H Different algorithms can be developed to solve the same problem. EK 4.1.2B Natural language and pseudocode describe algorithms so that humans can understand them. EK 4.2.1A Many problems can be solved in a reasonable time. EK 4.2.1B Reasonable time means that as the input size grows, the number of steps the algorithm takes is proportional to the square (or cube, fourth power, fifth power, etc.) of the size of the input. EK 4.2.1C Some problems cannot be solved in a reasonable time, even for small input sizes. EK 4.2.1D Some problems can be solved but not in a reasonable time. In these cases, heuristic approaches may be helpful to find solutions in		

reasonable time. EK 4.2.2A A heuristic is a technique that may allow us to find an approximate solution when typical methods fail to find an exact solution. EK 4.2.2B Heuristics may be helpful for finding an approximate solution more quickly when exact methods are too slow. (Exclusion statement: Specific heuristic solutions are beyond the scope of this course and the AP Exam). EK 4.2.2C Some optimization problems such as "find the best" or "find the smallest" cannot be solved in a reasonable time but approximations to the optimal solution can. EK 4.2.2D Some problems cannot be solved using any algorithm. EK 4.2.3A An undecidable problem may have instance that have an algorithmic solution, but there is no algorithmic solution that solves all instances of the problem EK 4.2.3B A decidable problem is one in which an algorithm can be constructed to answer "yes" or "no" for all inputs (e.g., "is the number even?"). EK 4.2.3C An undecidable problem is one in which no algorithm can be constructed that always leads to a correct yes-or-no answer. (Exclusion statement: Determining whether a given problem is undecidable is beyond the scope of this course and the AP Exam.) EK 4.2.4D Different correct algorithms for the same problem can have different efficiencies. EK 4.2.4H Linear search can be used when		

searching for an item in any list; binary search can be used only when the list is sorted. **<u>Big Idea 5 Programming:</u>** EK 5.1.2B Developing correct program components and then combining them helps in creating correct programs. EK 5.1.2C Incrementally adding tested program segments to correct working programs helps create large correct programs. EK 5.3.1D Procedures have names and may have parameters and return values. EK 5.3.1E Parameterization can generalize a specific solution. EK 5.3.1F Parameters generalize a solution by allowing a procedure to be used instead of duplicated code. EK 5.3.1G Parameters provide different values as input to procedures when they are called in a program. EK 5.3.1I Strings and string operations, including concatenation and some form of substring, are common in many programs. EK 5.3.1K Lists and list operations, such as add, remove, and search, are common in many programs. EK 5.3.1L Using lists and procedures as abstractions in programming can result in programs that are easier to develop and maintain.		

EK 5.5.1A Numbers and numerical concepts are fundamental to programming. EK 5.5.1G intuitive and formal reasoning about program components using Boolean concepts helps in developing correct programs. EK 5.5.1H Computational methods may use lists and collections to solve problems. EK 5.5.1I Lists and other collections can be treated as abstract data types (ADTs) in developing programs. 5.5.1J Basic operations on collections include adding elements, removing elements, iterating over all elements, and determining whether an element is in a collection. **<u>Big Idea 7 Global Impact:</u>** EK 7.1.1G Search trends are predictors. EK 7.1.1H Social media, such as blogs and Twitter, have enhanced dissemination. EK 7.1.2D Human capabilities are enhanced by digitally enabled collaboration. EK 7.1.2E Some online services use the contributions from many people to benefit both individuals and society. EK 7.1.2G The move from desktop computers to a proliferation of always-on mobile computers is leading to new applications. EK 7.3.1M Targeted advertising is used to help individuals, but it can be misused at both		

individual and aggregate levels. EK 7.5.1B Advanced search tools, Boolean logic, and key words can refine the search focus, and/or limit their searches based on a variety of factors (e.g., data, peer-review status, type of publication).		

Stage 2 – Evidence

Code	Evaluative Criteria	Assessment Evidence
M, T	M/C, Completion and Short Answer	Quiz on Unit 5--Students should score at least an 80% PERFORMANCE TASK(S):
A, M, T	Teacher Checklist	Quiz App Projects.The Quiz App presents a quiz about pioneers in computer science.The quiz uses lists, labels, images, and a button to create a simple question and answer quiz that displays 'Correct!' or 'Incorrect!' depending on the user's answer. An index variable is used to keep track of the current question and access its correct answer and corresponding image. The questions, answers, and corresponding images are in parallel lists where the first question in the question list matches with the first answer in the answer list and the first picture in the picture list, etc. Students code simple extensions of the existing Quiz App, including keeping a tally of correct/incorrect answers.
		OTHER EVIDENCE:
A, M	Exercises with feedback	AppInventorDrills • Refactoring Code -- coding drills that focus on ***procedures*** and ***procedures that return a result (functions)***. You are given an app that will require you to ***refactor*** its code, creating new procedures and functions.
A, M, T	Teacher Observation and Rubric	POGIL Activities: Search Algorithms and Limits of Algorithms
A, M, T	Teacher Observation	Blown to Bits Chapter 5 discussion

Stage 3 – Learning Plan		
Code	***Pre-Assessment*** Class discussion about sorting and searching algorithms. Ask about searching through an unordered list as opposed to an ordered list. Note if students recognize the efficiency gain. Ask students to sort an unordered pile of cards.	
	Summary of Key Learning Events and Instruction	Progress Monitoring
A, M	Teacher directs students to complete Logo 2 app. This version of Logo provides procedures with parameters. Problems include drawing polygons using procedures with one or more parameters. The lesson focuses on how procedures with parameters provide a more powerful abstraction for the forward(N) and turn(A) commands. Teacher circulates to provide assistance and enforces "ask three before me" rule.	App Completion Checklist
A, T	Search Algorithms. An introduction to search algorithms, including sequential (linear) search and binary search. Teacher leads students though various interactive guessing games that explain algorithms.. A POGIL activity shows students the binary guessing game and asks them to figure out the algorithm and express it in pseudocode.	Teacher observation during POGIL and interactive guessing game. Self check questions. Student reflection in portfolio.
A	Sorting Algorithms. An introduction to the problem of sorting with examples of bubble sort, merge sort, and bucket (radix) sort. Teacher demonstrates the algorithms using videos and card shuffling.	Self check questions. student reflection in portfolio.
A	Teacher directs students to complete Quiz App tutorial. A basic quiz app that uses parallel lists and indexing to keep track of questions and answers. Teacher circulates to provide assistance and enforces "ask three before me" rule.	App Completion Checklist
M, T	Analyzing Algorithms. Apps are used to experimentally analyze sorting and searching algorithms. By timing the algorithms on different sized lists and graphing the results, students can identify which algorithm is which just by the shape of its growth curve as *logarithmic (log_2 N), linear,* or *quadratic (N^2).*	Observation during app experiment. Students record, graph and analyze results. Self check questions. Student Reflection in website portfolio.

A, T	Limits of Algorithms. A video lecture introduces the concepts of *intractability* and *undecidability* and *heuristics* -- i.e., there are problems for which the best algorithms are incapable of solving the problem in a reasonable amount of time and there are problems which cannot be solved by means of an algorithm. POGIL activities focus on password protection (using intractable problem to protect a password) and Traveling Salesman Problem (using the nearest neighbor heuristic.)	Observation during POGIL. Self check questions. Student Reflection in website portfolio.
A, M, T	BB: Web Searches. Students read and students take turns leading a discussion on Chapter 5 of ***Blown to Bits***, which focuses on web searching and how searching is done.	Teacher observation during discussion. Student Reflection in website portfolio.

Unit 6: Communication through the Internet

<table>
<tr><th colspan="3">Stage 1 Desired Results</th></tr>
<tr><td rowspan="3">ESTABLISHED GOALS

Computational Thinking Practice 1: Connecting Computing
Identify impacts of computing.
Describe connections between people and computing.
Explain connections between computing concepts.

Computational Thinking Practice 2: Creating Computational Artifacts
Create a computational artifact with a practical, personal, or societal intent.
Select appropriate techniques to develop a computational artifact.
Use appropriate algorithms and information management principles.

<u>APCSP Learning Objectives:</u>

<u>Big Idea 1 Creativity:</u>

Learning Objective 1.2.2 Create a computational artifact using computing tools and techniques to solve a problem

<u>Big Idea 2 Abstraction:</u>

Learning Objective 2.2.1 Develop an abstraction when writing a program or creating other computational artifacts.</td><td colspan="2">Transfer
Students will be able to independently use their learning to…

Measure the performance of an internet-enabled device.

Work collaboratively and independently to solve problems by applying the problem solving process.</td></tr>
<tr><td colspan="2">Meaning</td></tr>
<tr><td>UNDERSTANDINGS
Students will understand that…

APCSP Big Idea 1: Creativity.
1.2 Computing enables people to use creative development processes when using computing tools and techniques to create computational artifacts for creative expression of ideas or to solve a problem.

APCSP Big Idea 5: Programming
5.3 Programming is facilitated by appropriate abstractions.

APCSP Big Idea 6: The Internet
6.1 The Internet is a network of autonomous systems.
6.2 Characteristics of the Internet influence the systems built on it.
6.3 Cybersecurity is an important concern for the Internet and the systems built on it.

APCSP Big Idea 7: Global Impact
7.1 Computing enhances communication, interaction, and cognition.</td><td>ESSENTIAL QUESTIONS
Students will keep considering…

What is the Internet? How is it built? How does it function?

What aspects of the Internet's design and development have helped it scale and flourish?

How is cybersecurity impacting the ever increasing number of Internet users?

How does computing enhance communication, interaction, and cognition?

How does computing enable innovation?

What are some potential beneficial and harmful effects of computing?

How do economic, social, and cultural contexts influence innovation and the use of computing?</td></tr>
</table>

<table>
<tr>
<td rowspan="4">Learning Objective 2.2.2 Use multiple levels of abstraction to write programs
Learning Objective 2.2.3 Identify multiple levels of abstractions that are used when writing programs.

<u>Big Idea 5 Programming:</u>

Learning Objective 5.2.1 Explain how programs implement algorithms.

<u>Big Idea 6 The Internet:</u>

Learning Objective 6.1.1 Explain the abstractions in the Internet and how the Internet functions. (Exclusion statement: Specific devices used to implement the abstractions in the Internet are beyond the scope of this course and the AP Exam.)

Learning Objective 6.2.1 Explain characteristics of the Internet and the systems built on it.

Learning Objective 6.2.2 Explain how the characteristics of the Internet influence the systems built on it.

Learning Objective 6.3.1 Identify existing cybersecurity concerns and potential options to address these issues with the Internet and the systems built on it.

<u>Big Idea 7 Global Impact:</u>
Learning Objective 7.1.1 Explain how computing innovations affect communication, interaction, and cognition.</td>
<td>7.3 Computing has a global affect - both beneficial and harmful - on people and society.
7.4 Computing innovations influence and are influenced by the economic, social, and cultural contexts in which they are designed and used.</td>
<td></td>
</tr>
<tr>
<td colspan="2">Acquisition</td>
</tr>
<tr>
<td>Students will know…</td>
<td>Students will be skilled at…</td>
</tr>
<tr>
<td>
AES: advanced encryption standard, a symmetric 128-bit block data encryption technique
certification authority: (CA), a trusted organization or company that issues digital certificates used to create digital signatures and public-private key pairs
cipher text: data that has been encrypted
DES: data encryption standard, a popular symmetric-key encryption method that uses a 56-bit key and uses a block cipher method which breaks text into 64-bit blocks and then encrypts them
decryption: the process of decoding data that has been encrypted into a secret format
encryption: the translation of data into secret code
packet: a piece of message transmitted over a packet-switching network
plain text: refers to textual data in ASCII format. Plain text is the most portable format because it is supported by nearly every application on every machine
router: a device that forwards data packets along networks. A router is connected to at least two networks are located at gateways
</td>
<td>
Using the Internet Society's Interactive map of global internet statistics to find data on a specified topic.
Using the http://www.bandwidthplace.com/ tool (or http://speedtest.xfinity.com/) to measure the bandwidth and latency of an Internet connection.
Explaining how latency differs from bandwidth
Applying the ping and trace tools to determine the following:

Did any of the servers lose packets or time out? Some servers will block ping and trace for security reasons which are seen as time outs. Were there any surprising locations in the hops that the packet went through?
Did different trials have different results for the same destination? Do packets always get routed in the same way?
Can you guess where the network-tools server is located based on the latency data you collected? Can you confirm your guess using trace or whois (which gives you

</td>
</tr>
</table>

Learning Objective 7.3.1 Analyze the beneficial and harmful effects of computing. Learning Objective 7.4.1 Explain the connections between computing and economic, social and cultural context **<u>APCSP Essential Knowledge:</u>** **<u>Big Idea 2 Abstraction:</u>** EK 2.2.1C An abstraction generalizes functionality with input parameters that allow software reuse.,(Exclusion statement: An understanding of the difference between value and reference parameters is beyond the scope of this course and the AP Exam. **<u>Big Idea 4 Algorithms:</u>** EK 4.1.1C Selection uses a Boolean condition to determine which of two parts of an algorithm is used. **<u>Big Idea 5 Programming:</u>** EK 5.1.2C Incrementally adding tested program segments to correct working programs helps create large correct programs. EK 5.1.2D Program documentation helps programmers develop and maintain correct programs to efficiently solve problems. EK 5.1.2E Documentation about program components, such as blocks and procedures, helps in developing and maintaining programs. EK 5.2.1G A process may execute by itself or with other processes.	● A network is a group of two or more computers that are linked together. ● The World Wide Web is an Internet application that is based on the HTTP protocol. ● A client is a computer or software application that requests services from a server located on the internet -- e.g., a Web browser is an example of a client. ● SMTP/POP Simple Mail Transfer Protocol (SMTP) and Post Office Protocol (POP) are sets of rules that govern the email services. ● URI Uniform Resource Identifier (URI) is WWW identifier that uniquely identifies a resource on the WWW -- e.g., http://host.com' ● A protocol is a system of rules that govern the behavior of some system. ● A modem is a device that connects a computer to an Internet Service Provider (ISP ● The Digital Divide is the gap between those who have access to the Internet and computers and those who do not ● An ethernet is a network that uses wires to connect computers ● A host is a computer that's connected directly to the Internet -- often a computer that provides certain services or resources. ● Bandwidth is the rate at which data is downloaded or uploaded in a network ● A LAN (local area network) connects computers within a school or home ● A WAN (wide area network) connects devices over a broad geographic region -- e.g., a telephone network. ● A Server is a host that provides some kind of service -- e.g., Google's Gmail service ● HTTP HyperText Transfer Protocol is the set of rules that governs the WWW application.	information about who owns a server)? ○ How does the number of hops in the trace affect latency (the round trip time seen in ping)? ○ How does geographical distance affect latency? What are some other factors that may be affecting latency? ● Explaining how the geographical distance between the source and destination hosts on a network affects latency ● Explaining the benefits of packet switching. ● Explain how DNS names and IP addresses are assigned. ● Explain how messages are queued and broken into packets ● Building an app that implements Caesar cipher encryption and decryption, ● Using local variables (as opposed to global variables) in an app, ● Using a function (a procedure that produces a value) in an app. ● Writing an an enhancement for a function based on specifications ● Use existing procedures in another procedure ● Explain how public key encryption is not symmetric. ● Explain why is it necessary to use open standards in cryptography? ● Applying debugging strategies to existing programs to fix defects in them ● Applying good coding practices when writing new code to prevent defects ● Documenting code to make it easier to read and understand by other programmers. ● Identify examples of syntax errors ● Identify examples of semantic errors ● Explaining how encryption and cryptography

EK 5.2.1H A process may execute on one or several CPUs. EK 5.3.1A Procedures are reusable programming abstractions. EK 5.3.1B A procedure is a named grouping of programming instructions. EK 5.3.1C Procedures reduce the complexity of writing and maintaining programs. EK 5.3.1D Procedures have names and may have parameters and return values. EK 5.3.1G Parameters provide different values as input to procedures when they are called in a program. EK 5.4.1E Locating and correcting errors in a program is called debugging the program. EK 5.4.1F Knowledge of what a program is supposed to do is required in order to find most program errors. EK 5.4.1H Visual displays (or different modalities) of a program state can help in finding errors. EK 5.4.1K Correctness of a program depends on correctness of program components, including code blocks and procedures. **<u>Big Idea 6 The Internet:</u>** EK 6.1.1A The Internet connects devices and networks all over the world.	● HTML HyperText Markup Language is a language for formatting Web pages. ● A router is a device that transmits data between two different networks ● An Internet Service Provider (ISP) is a company that provides customers with Internet access. ● Wifi is a network that uses radio waves to connect devices (computers, smart phones, printers). ● Latency is a measure of the time it takes for a piece of data to reach its destination ● In the client/server model algorithms are executed by cooperating computers (CPUs) located on the internet. ● There are four layers in the tcp/ip model: ○ Application Layer (protocol: SMTP) ■ Send: Composes a message to another student and passes it to the Transport layer. ■ Receive: Receives and reads out messages from other students passed on from the Transport layer. ○ Transport Layer (protocol: TCP) ■ Send: Splits the message into packets, adds TCP headers to number the packets, and sends them to the Internet layer. ■ Receive: Receives packets from the Internet layer, puts them in order, and passes them to the application layer when all is received. ○ Internet Layer (protocol: IP) ■ Send: Uses a routing table (given in the handout) to add the destination IP address to each packet and passes them to the Link layer. ■ Receive: Receives packets from the Link layer and checks that it's their	are related to cybersecurity ● Analyzing the main legal and ethical reasons for/against letting the government have a back door

EK 6.1.1B An end-to-end architecture facilitates connecting new devices and networks on the internet. EK 6.1.1C Devices and networks that make up the Internet are connected and communicate using addresses and protocols. EK 6.1.1D The Internet and the systems built on it facilitate collaboration. EK 6.1.1E Connecting new devices to the Internet is enabled by the assignment of an Internet protocol (IP) address. EK 6.1.1F The Internet is built on evolving standards, including those for addresses and names. (Exclusion statement: Specific details of any particular standard for addresses are beyond the scope of this course and the AP Exam.) EK 6.1.1G The domain name system (DNS) translates names to IP addresses. EK 6.1.1H The number of devices that could use an IP address has grown so fast that a new protocol (IPv6) has been established to handle routing of many more devices. EK 6.1.1I Standards such as hypertext transfer protocol (HTTP), IP, and simple mail transfer protocol (SMTP) are developed and overseen by the Internet Engineering Task Force (IETF). EK 6.2.1A The Internet and the systems built on it are hierarchical and redundant. EK 6.2.1B The domain name syntax is hierarchical.	own group's IP address. If it is, it passes it to the transport layer. If it is not, it gives it back to the link layer to give to another group. ○ Link Layer (protocol: Ethernet) ■ Send: Passes the individual packets randomly to the link layer of other groups. ■ Receive: Receives packets from other groups and passes them to the Internet Layer. ● Ping is a networking utility used by network administrators to test the reachability of a host on the Internet ● Traceroute is a networking utility used to trace the route and measure delays of packets moving through the Internet. ● a packet is a collection of data used by the TCP/IP protocol to transmit data across the Internet. Each packet contains routing data as well as the content of the message. ● packet switching is the method by which information is transmitted through the Internet. Information is broken into packets and each packet is routed independently from source to destination. ● The Internet is organized into several abstraction layers that are controlled by various protocols. From the bottom up, we have the link layer (Ethernet protocol), the Internet layer (IP), transport layer (TCP), and application layer (HTTP). ● An Internet domain name is a hierarchical name (such as trincoll.edu) that identifies an domain and an institution on the Internet. Top level domains include com, edu, gov ● Domain Name System (or Service or Server) is an Internet service that translates domain names into IP addresses.	

EK 6.2.1C IP addresses are hierarchical. EK 6.2.1D Routing on the Internet is fault tolerant and redundant. EK 6.2.2A Hierarchy and redundancy helps systems scale. EK 6.2.2B The redundancy of routing (i.e., more than one way to route data) between two points on the Internet increases the reliability of the Internet and helps it scale to more devices and more people. EK 6.2.2C Hierarchy in the DNS helps that system scale EK 6.2.2D Interfaces and protocols enable widespread use of the Internet. EK 6.2.2E Open standards fuel the growth of the Internet. EK 6.2.2F The internet is a packet-switched system through which digital data is sent by breaking the data into blocks of bits called packets, which contain both the data being transmitted and control information for routing data. (Exclusion statement: Specific details of any particular packet-switching system are beyond the scope of this course and the AP Exam). EK 6.2.2G Standards for packets and routing include transmission control protocol/Internet protocol (TCP/IP). (Exclusion statement: Specific technical details of how TCP/IP works are beyond the scope for this course and the AP Exam). EK 6.2.2H Standards for sharing information and	• A cipher is a system for creating secret messages. • Cryptography means, literally, 'secret writing'. The art and science of writing secret messages. • Encryption is the process of using a secret key to convert plaintext into ciphertext. • Plaintext is the unencrypted, readable message • Ciphertext is an unreadable, secret message. • Decryption is the process of using a secret key to convert ciphertext into plaintext. • The encryption key is a piece of secret data used in by encryption and decryption algorithms • The encryption algorithm uses a secret key to encrypt messages. • In a substitution cipher letters from a ciphertext alphabet are substituted for the letters in a plaintext message in a systematic way. • The caesar cipher is an example of a substitution cipher. • A function is a procedure that returns a value. • A local variable (in contrast to a global variable) is one that has a limited scope, which means that it only exists and can only be used within a block of code, for example in a procedure or a function • In a symmetric encryption system the same key is used for both encryption and decryption. • In a transposition cipher letters in the plaintext are rearranged without substitution. • In cryptography, a brute force attack attempts to try every possible encryption key to break a secret message. • Frequency analysis counts the occurrence of the letters in an encrypted message in an	

communicating between browsers and servers on the Web include HTTP and secure sockets layer/transport layer security (SSL/TLS). (Exclusion statement: Understanding the technical aspects of how SSL/TLS works is beyond the scope of this course and the AP Exam.) EK 6.2.2I The size and speed of systems affect their use. EK 6.2.2J The bandwidth of a system is a measure of bit rate - the amount of data (measured in bits) that can be sent in a fixed amount of time. EK 6.2.2K The latency of a system is the time elapsed between the transmission and the receipt of a request. EK 6.3.1A The trust model of the Internet involves trade-offs. EK 6.3.1B The DNS was not designed to be completely secure. EK 6.3.1C Implementing cybersecurity has software, hardware, and human components. EK 6.3.1F Phishing, viruses, and other attacks have human and software components. EK 6.3.1G Antivirus software and firewalls can help prevent unauthorized access to private data. EK 6.3.1H Cryptography is essential to many models of cybersecurity. EK 6.3.1I Cryptography has a mathematical foundation. (Exclusion statement: Specific mathematical functions used in cryptography are	effort to discover patterns that might reveal the encryption key ● In a polyalphabetic substitution system multiple alphabets are used to encrypt a single message. ● The one time pad system is an example of perfect (unbreakable) encryption, which is achieved by using, only once, a random polyalphabetic key that is as long the message itself. ● In cryptography, the key exchange problem is the problem of sharing a secret key between Alice and Bob, without Eve, an eavesdropper, being able to intercept it. ● An asymmetric cipher is one in which separate but related keys are used for encryption and decryption. ● Public key cryptography is a cryptographic system that uses two keys -- a public key known to everyone and a private or secret key known only to the recipient of the message. When Bob wants to send a secure message to Alice, he uses Alice's public key to encrypt the message. Alice then uses her private key to decrypt it. ● Diffie Hellman is an algorithm used to establish a shared secret between two parties. It is primarily used to exchange a symmetric cryptographic key among two parties, Alice and Bob, who wish to communicate securely ● Rivest-Shamir-Adleman (RSA) is a cryptosystem for public-key encryption, and is widely used for securing sensitive data, particularly when being sent over an insecure network such as the Internet. ● HTTPS is a protocol for secure (trusted, encrypted) communication over the Internet. ● SSL (Secure Socket Layer) is a protocol for	

beyond the scope of this course and the AP Exam.) EK 6.3.1J Open standards help ensure cryptography is secure. EK 6.3.1K Symmetric encryption is a method of encryption involving one key for encryption and decryption. (Exclusion statement: The methods used in encryption are beyond the scope of this course and the AP Exam). **<u>Big Idea 7 Global Impact:</u>** EK 7.3.1A Innovations enabled by computing raise legal and ethical concerns. EK 7.3.1D Both authenticated and anonymous access to digital information raise legal and ethical concerns. EK 7.3.1E Commercial and governmental censorship of digital information raise legal and ethical concerns. EK 7.3.1G Privacy and security concerns arise in the development and use of computational systems and artifacts. EK 7.3.1J Technology enables the collection, use, and exploitation of information about, by, and for individuals, groups, and institutions. EK 7.3.1I Anonymity in online interactions can be enabled through the use of online anonymity software and proxy servers. EK 7.4.1D Groups and individuals are affected by the "digital divide" - differing access to computing	establishing an encrypted link between a web server and a browser. ● In cryptography, a certificate authority (CA) is an entity that issues digital certificates ● A digital certificate is a data packet that certifies the holder of a public key ● A trust model is the use of a trusted third party to verify the trustworthiness of a digital certificate ● Debugging is the process of removing errors from computer hardware or software. ● A computer bug is an informal term for error in computer hardware or software -- the term was coined by Grace Hopper ● A syntax error is an error that results from a violation of the programming language grammatical rules. ● A semantic error is an error in which the program is not working as it is designed to work	

and the Internet based on socioeconomic or geographic characteristics.		

Stage 2 – Evidence		
Code	**Evaluative Criteria**	**Assessment Evidence**
M, T	M/C, Completion, Short Answer	Quiz on Unit 6: Students should score at least an 80% PERFORMANCE TASK(S):
M, T	College Board Rubric	EXPLORE PT 2 is the official EXPLORE impact of a computing innovations performance task that will be submitted to the College Board. The EXPLORE task is one of two required performance tasks by the College Board - a programming one (CREATE) and a written one (EXPLORE). In this written performance task, students work independently to research a computing innovation of their choosing that has had significant impact (both positive and negative) on our society. This includes finding credible, reliable, and recent sources, as well as, answering a series of prompts about their chosen innovation. Students then create a visual artifact that demonstrates what they learned about one or more of the effects of the innovation.
		OTHER EVIDENCE:
M	Correct/Incorrect answers with feedback	Coding Drills: String processing -- coding drills that focus on processing text data using built-in functions and loops.
A, M, T	Teacher Observation and Rubric	POGIL Activities: IP Addresses and Domain Names
M, T	Teacher Observation	Blown to Bits Chapter 5 discussion

Stage 3 – Learning Plan		
Code	***Pre-Assessment*** Think, pair shares on concepts related to the Internet, Cryptography, DNS and IP addresses.	
	Summary of Key Learning Events and Instruction	Progress Monitoring
A, T	Internet: Basic Concepts and Terminology. Teacher shows a 3-part video lecture that describes what the Internet is, how it differs from the World Wide Web, and how its performance is measured. Teacher demonstrates the use of various online tools to measure *latency* and *bandwidth.* Students use these tools in an analysis activity.	Teacher observation during activity. Self check questions and student reflection in online portfolio.
M, T	Teacher directs students to resources on Internet Architecture and Packet Switching. This lesson goes more deeply into the infrastructure and mechanics of the Internet. It explains *packet switching*, *TCP/IP* and the protocol hierarchy. Students break into POGIL groups to use PING and traceroute to investigate how information travels on the internet.	Teacher observation during POGIL. Self check questions and student reflection in online portfolio.
A, M, T	IP Addresses and Domain Names. Teacher demonstrates use of a DNS simulator app. students use a DNS simulator app to send messages to other clients on a router. Students break into POGIL groups to learn about DNS, IP addresses, and packets.	Teacher observation during DNS activity and POGIL. Self check questions and student reflection in online portfolio.
A	Cryptography Basics. Introduction to cryptography (secret writing). This lesson focuses on classical cryptography, including Caesar cipher, substitution cipher, transposition cipher, Vigenere cipher, and frequency analysis. It ends with the *key exchange problem.* Students participate in inquiry activities which include using interactive tools to encrypt, decrypt, and analyze secret messages.	Teacher observation during cryptography activities. Self check questions and student reflection in online portfolio.
M, T	Teacher directs students to complete Caesar Cipher App tutorial. this app helps students understand and implement the cryptography concepts learned in the previous lesson. Teacher circulates to provide assistance and enforces “ask three before me” rule.	App Completion checklist.

A	Cryptography: Securing the Internet. Teacher plays videos that introduce the Diffie-Hellman key exchange algorithm and *public key cryptography (PKC).* The videos demonstrate how PKC is used to implement secure transactions over the Internet. Students complete activities which include interactive public key encryptions.	Teacher observation during cryptography activities. Self check questions and student reflection in online portfolio.
M, T	Debugging Caesar Cipher. Teacher directs students to work in pairs to find and fix several errors contained in the Caesar Cipher app.	Error checklist.
A, M, T	BB: Cryptography and the Government. Students read and lead a discussion of small sections of Chapter 5 of *Blown to Bits*, which focuses on encryption and how it is used to secure transactions on the Internet. Students read a short Wikipedia on recent the Apple vs. FBI controversy.	Teacher observation during discussion, self check questions and student reflection in online portfolio.

Unit 7: Using and Analyzing Data and Information

<table>
<tr><th colspan="3">Stage 1 Desired Results</th></tr>
<tr><td rowspan="3">ESTABLISHED GOALS

Computational Thinking Practice 1: Connecting Computing
Identify impacts of computing.
Describe connections between people and computing.
Explain connections between computing concepts.

Computational Thinking Practice 2: Creating Computational Artifacts
Create a computational artifact with a practical, personal, or societal intent.
Select appropriate techniques to develop a computational artifact.
Use appropriate algorithms and information management principles.

Computational Thinking Practice 3: Abstracting
Explain how data, information, or knowledge is represented for computational use.
Explain how abstractions are used in computation or modeling.
Identify abstractions
Describe modeling in a computational context.

Computational Thinking Practice 4: Analyzing Problems and Artifacts</td><td colspan="2">Transfer

Students will be able to independently use their learning to…

Perform research to identify a large data set.

Use Google Sheets and Google Maps to analyze a large data set.</td></tr>
<tr><td colspan="2">Meaning</td></tr>
<tr><td>UNDERSTANDINGS
Students will understand that…

APCSP Big Idea 1: Creativity.
1.2 Computing enables people to use creative development processes when using computing tools and techniques to create computational artifacts for creative expression of ideas or to solve a problem.

APCSP Big Idea 3: Data and Information
3.1 People use computer programs to process information to gain insight and knowledge.
3.2 Computing facilitates exploration and the discovery of connections in information.
3.3 There are trade-offs when representing information as digital data.

APCSP Big Idea 5: Programming
5.1 Programs can be developed to solve problems (to help people, organizations or society); for creative expression; to satisfy personal curiosity or to create new knowledge.
5.2 People write programs to execute algorithms.</td><td>ESSENTIAL QUESTIONS
Students will keep considering…

How can computation be employed to help people process data and information to gain insight and knowledge?

How can computation be employed to facilitate exploration and discovery when working with data?

What considerations and trade-offs arise in the computational manipulation of data?

What opportunities do large data sets provide for solving problems and creating knowledge?</td></tr>
</table>

<table>
<tr>
<td rowspan="3">Evaluate a proposed solution to a problem.
Locate and correct errors
Explain how an artifact functions
Justify appropriateness and correctness of a solution, model, or artifact.

<u>APCSP Learning Objectives:</u>

<u>Big Idea 1 Creativity:</u>

Learning Objective 1.2.1 Create a computational artifact for creative expression.
Learning Objective 1.2.2 Create a computational artifact using computing tools and techniques to solve a problem

<u>Big Idea 3 Data and Information:</u>

Learning Objective 3.1.1 Use computers to process information, find patterns, and test hypotheses about digitally processed information to gain insight and knowledge.
Learning Objective 3.1.2 Collaborate when processing information to gain insight and knowledge

Learning Objective 3.1.3 Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language.

Learning Objective 3.2.1 Extract information from data to discover and explain connections, patterns, or trends.

Learning Objective 3.2.2 Use large data sets to explore and discover information and knowledge.

Learning Objective 3.3.1 Analyze how data</td>
<td>5.3 Programming is facilitated by appropriate abstractions.
5.5 Programming uses mathematical and logical concepts.

APCSP Big Idea 7: Global Impact
7.1 Computing enhances communication, interaction, and cognition.
7.2 Computing enables innovation in nearly every field.
7.3 Computing has a global affect - both beneficial and harmful - on people and society.</td>
<td></td>
</tr>
<tr>
<td colspan="2">Acquisition</td>
</tr>
<tr>
<td>Students will know…

● centralized network: when the resources and workload are coordinated and managed by a centralized computer (server)
● centralized systems: collect files at a central computer for people to download
● commons: a system of sharing that minimizes the need for fine-grained property restrictions
● copyright: a legal right that grants the creator of an original work exclusive rights for its use and distribution
● creative commons: a set of licenses that allow creators to communicate which rights they reserve, and which rights they waive for the benefit of recipients or other creators
● decentralized network: when the allocation of resources and workload are distributed to individual devices on a network
● DMCA - Digital Millennium Copyright Act: US copyright law that criminalizes production and</td>
<td>Students will be skilled at…

● Given a data set, describing the data it contains, and creating a hypothesis about the data.
● Explaining the purpose of a data visualization
● Creating an app that can be used to poll individuals and store responses on the web
● Explaining the concept of centralizing and sharing data using a database
● Explaining the difference between synchronous and asynchronous operations
● Explaining how a database abstraction reduces detail in a program
● Describing the disadvantages of using a public database.
● Giving an example of a race condition
● Interpreting data visualizations
● Manipulating data in a Google Sheet
● Adding charts to a google sheets, making hypotheses and drawing conclusions from the</td>
</tr>
</table>

representation, storage, security, and transmission of data involve computational manipulation of information. **<u>Big Idea 5 Programming:</u>** Learning Objective 5.1.2 Develop a correct program to solve problems. Learning Objective 5.3.1 Use abstraction to manage complexity in programs. Learning Objective 5.5.1 Employ appropriate mathematical and logical concepts in programming **<u>Big Idea 7 Global Impact:</u>** Learning Objective 7.1.1 Explain how computing innovations affect communication, interaction, and cognition. Learning Objective 7.2.1 Explain how computing has impacted innovations in other fields. Learning Objective 7.3.1 Analyze the beneficial and harmful effects of computing. **<u>APCSP Essential Knowledge:</u>** **<u>Big Idea 3 Data and Information:</u>** 3.1.1A Computers are used in an iterative and interactive way when processing digital information to gain insight and knowledge. 3.1.1B Digital information can be filtered and cleaned by using computers to process information. 3.1.1C Combining data sources, clustering data, and data classification are part of the process of using computers to process information. 3.1.1D Insight and knowledge can be obtained	dissemination of technology, devices, or services intended to circumvent measures that control access to copyrighted works ● DRAM: dynamic random access memory ● DRM - digital rights management: various access control technologies that are used to restrict usage of proprietary hardware and copyrighted works ● flooding: each computer in a file-sharing network maintains a list of other computers in the network. ● fair use: limited use of copyrighted material without having to first get permission from the copyright holder ● gigabyte: 1,024 megabytes or 1,073,741,824 bytes ● open source: unrestricted access and unrestricted reuse ● peer-to-peer: a distributed architecture or network that divides tasks between peers, each of which participate in the application ● peer-to-peer architecture: a type of network in which each workstation has equivalent capabilities and responsibilities ● piracy: the unauthorized use or reproduction of another's work ● sealed storage: an application that lets you encrypt files in such a way that they can be decrypted only on particular computers that you specify. ● TPM: trusted platform module ● Big data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications.	data in the charts. ● Using filtering tools to explore a data set and look at patterns. ● Exporting data from a Google Sheet into a Google map. ● Interpreting data in a spreadsheet ● Writing spreadsheet formulas ● Using data visualizations to answer questions ● Working collaboratively to research, investigate and analyze a large data set. ● Brainstorming hypotheses that can be answered by data in a large data set. ● Creating unique data visualizations that illustrate hypotheses ● Use the tools available on the Google Spreadsheets and Google My Maps ● Identify copyrighted material and proper use. ● Identify peer-to-peer networks. ● Describe open access and creative commons and how they allow the legal sharing of digital information.

from translating and transforming digitally represented information. 3.1.1E Patterns can emerge when data is transformed using computational tools. 3.1.2C Communication between participants working on data-driven problems gives rise to enhanced insights and knowledge. 3.1.2E Collaborating face-to-face and using online collaborative tools can facilitate processing information to gain insight and knowledge. 3.1.3A Visualization tools and software can communicate information about data. 3.1.3B Tables, diagrams, and textual displays can be used in communicating insight and knowledge gained from data. 3.1.3C Summaries of data analyzed computationally can be effective in communicating insight and knowledge gained from digitally represented information. 3.1.3D Transforming information can be effective in communicating knowledge gained from data. 3.1.3E Interactivity with data is an aspect of communicating. 3.2.1A Large data sets provide opportunities and challenges for extracting information and knowledge. 3.2.1B Large data sets provide opportunities for identifying trends, making connections in data, and solving problems.		

3.2.1C Computing tools facilitate the discovery of connections in information within large data sets. 3.2.1E Information filtering systems are important tools for finding information and recognizing patterns in the information. 3.2.1F Software tools, including spreadsheets and databases, help to efficiently organize and find trends in information. (Exclusion statement: Students are not expected to know specific formulas or options available in spreadsheet or database software packages). 3.2.2A Large data sets include data such as transactions, measurements, texts, sounds, images, and videos. 3.2.2B The storing, processing, and curating of large data sets is challenging. 3.2.2C Structuring large data sets for analysis can be challenging. 3.2.2D Maintaining privacy of large data sets containing personal information can be challenging. 3.2.2E Scalability of systems is an important consideration when data sets are large. 3.2.2F The size or scale of a system that stores data affects how that data set is used. 3.3.1G Data is stored in many formats depending on its characteristics (e.g., size and intended use). 3.2.2H Analytical techniques to store, manage,		

transmit, and process data sets change as the size of data sets scale. **<u>Big Idea 5 Programming:</u>** 5.3.1H Data abstraction provides a means of separating behavior from implementation. **<u>Big Idea 7 Global Impact:</u>** 7.1.1E Widespread access to information facilitates the identification of problems, development of solutions, and dissemination of results. 7.1.1F Public data provides widespread access and enables solutions to identified problems. 7.2.1A Machine learning and data mining have enabled innovation in medicine, business, and science. 7.2.1B Scientific computing has enabled innovation in science and business. 7.2.1C Computing enables innovation by providing access to and sharing of information. 7.2.1D Open access and Creative Commons have enabled broad access to digital information. 7.2.1E Open and curated scientific databases have benefited scientific researchers. 7.2.1G Advances in computing as an enabling technology have generated and increased the creativity in other fields. 7.3.1A Innovations enabled by computing raise legal and ethical concerns.		

7.3.1B Commercial access to music and movie downloads and streaming raises legal and ethical concerns. 7.3.1C Access to digital content via peer-to-peer networks raises legal and ethical concerns. 7.3.1F Open source and licensing of software and content raise legal and ethical concerns. 7.3.1N Widespread access to digitized information raises questions about intellectual property. 7.3.1O Creation of digital audio, video, and textual content by combining existing content has been impacted by copyright concerns. 7.3.1P The Digital Millennium Copyright Act (DMCA) has been a benefit and a challenge in making copyrighted digital material widely available. 7.3.1Q Open source and free software have practical, business, and ethical impacts on widespread access to programs, libraries, and code.		

Stage 2 – Evidence		
Code	**Evaluative Criteria**	**Assessment Evidence**
M	M/C, Completion and Short Answer	Unit 7 Quiz (Students should score an 80%) PERFORMANCE TASK(S):
M, T	Teacher Rubric	Data Visualization Project Students work in pairs to identify a large data set that interests them, then formulate hypotheses and analyze the data using Google Sheets to shed light on the hypotheses.
		OTHER EVIDENCE:
A, M, T	Teacher Observation	Blown to Bits Chapter 6 discussion

Stage 3 – Learning Plan		
Code	***Pre-Assessment*** Skill activity where students are given a set of data entry and analysis tasks in Google Sheets.	
	Summary of Key Learning Events and Instruction	Progress Monitoring
A, T	Big Data. Teacher plays a set of lectures describing the scope and the challenges involved in managing massive data sets. Includes description of the *Map Reduce* algorithm. Students choose a data set to analyze during activity in preparation for upcoming project.	Teacher observes and gives feedback on data set and hypothesis during activity. Self Check Questions. Student reflection in portfolio.
A	Teacher directs students to complete Clicker App with TinyWebDB tutorial. Teacher circulates to provide assistance and enforces "ask three before me" rule.	App Completion checklist.
A, M, T	Teacher directs students to complete the Clicker App with Firebase tutorial This tutorial uses the FirebaseDB component to store data to the Cloud. The concept of an asynchronous process is introduced to explain how a Web service works. A nested if/else algorithm is used to process requests. The last part of the lesson shows how to store images on the Web, with their URLs stored in FirebaseDb. Teacher circulates to provide assistance and enforces "ask three before me" rule.	App Completion checklist.
A, T	Visualizing Data. Teacher directs students to complete a sequence of activities that use Google sheets and Google Maps to process and visualize a data set.	Teacher observation during activity. Self Check Questions. Student reflection in portfolio.
M, T	Data Visualization Project Students work in pairs to identify a large data set that interests them, then formulate hypotheses and analyze the data to shed light on the hypotheses.	Teacher observation and feedback during activity.
A, M, T	BB: Who Owns the Bits. Students read and lead a discussion on Chapter 6 of *Blown to Bits*, which focuses on the issue of copyright.	Teacher observation during discussion, self check questions and student reflection in online portfolio.

Unit 8: APCS Principles Exam Prep

<table>
<tr><th colspan="3">Stage 1 Desired Results</th></tr>
<tr><td rowspan="3">ESTABLISHED GOALS

Computational Thinking Practice 1: Connecting Computing
Identify impacts of computing.
Describe connections between people and computing.
Explain connections between computing concepts.

Computational Thinking Practice 2: Creating Computational Artifacts
Create a computational artifact with a practical, personal, or societal intent.
Select appropriate techniques to develop a computational artifact.
Use appropriate algorithms and information management principles.

Computational Thinking Practice 3: Abstracting
Explain how data, information, or knowledge is represented for computational use.
Explain how abstractions are used in computation or modeling.
Identify abstractions
Describe modeling in a computational context.

Computational Thinking Practice 4: Analyzing Problems and Artifacts</td><td colspan="2">Transfer

Students will be able to independently use their learning to…

Employ strategies to effectively answer questions on a standardized test.

Read questions for meaning, looking for key words (only, every, etc).</td></tr>
<tr><td colspan="2">Meaning</td></tr>
<tr><td>UNDERSTANDINGS
Students will understand that…

APCSP Big Idea 1: Creativity.
1.2 Computing enables people to use creative development processes when using computing tools and techniques to create computational artifacts for creative expression of ideas or to solve a problem.

APCSP Big Idea 3: Data and Information
3.1 People use computer programs to process information to gain insight and knowledge.
3.2 Computing facilitates exploration and the discovery of connections in information.
3.3 There are trade-offs when representing information as digital data.

APCSP Big Idea 5: Programming
5.1 Programs can be developed to solve problems (to help people, organizations or society); for creative expression; to satisfy personal curiosity or to create new knowledge.
5.2 People write programs to execute algorithms.
5.3 Programming is facilitated by appropriate</td><td>ESSENTIAL QUESTIONS
Students will keep considering…</td></tr>
</table>

<table>
<tr>
<td rowspan="3">Evaluate a proposed solution to a problem.
Locate and correct errors
Explain how an artifact functions
Justify appropriateness and correctness of a solution, model, or artifact.

<u>APCSP Learning Objectives:</u>

<u>Big Idea 1 Creativity:</u>

Learning Objective 1.2.1 Create a computational artifact for creative expression.
Learning Objective 1.2.2 Create a computational artifact using computing tools and techniques to solve a problem

<u>Big Idea 3 Data and Information:</u>

Learning Objective 3.1.1 Use computers to process information, find patterns, and test hypotheses about digitally processed information to gain insight and knowledge.

Learning Objective 3.1.2 Collaborate when processing information to gain insight and knowledge

Learning Objective 3.1.3 Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language.

Learning Objective 3.2.1 Extract information from data to discover and explain connections, patterns, or trends.

Learning Objective 3.2.2 Use large data sets to explore and discover information and knowledge.</td>
<td>abstractions.
5.5 Programming uses mathematical and logical concepts.

APCSP Big Idea 7: Global Impact
7.1 Computing enhances communication, interaction, and cognition.
7.2 Computing enables innovation in nearly every field.
7.3 Computing has a global affect - both beneficial and harmful - on people and society.</td>
<td></td>
</tr>
<tr>
<td colspan="2">Acquisition</td>
</tr>
<tr>
<td>Students will know…

● The college board tests do not penalize for incorrect answers.
● The exam is 120 minutes (2 hours) long
● There are approximately 74 multiple-choice questions
● There is no designated programming language
● The exam is taken using paper and pencil
● On the exam, you will see two types of multiple-choice questions:
○ Single-Select Multiple-Choice: Students select 1 answer from among 4 options.
○ Multiple-Select Multiple-Choice: Students select 2 answers from among 4 options</td>
<td>Students will be skilled at…

● Applying strategies for taking multiple choice tests such as eliminating answers and skimming the question to determine whether it can be answered easily or marked to return to.
● Decipher and answer questions regarding pseudocode effectively.
● Monitoring the time during a multiple choice test to effectively answer the highest number of questions correctly.
● Read multiple choice questions carefully, looking for key words that change the meaning of the question.
● Apply the software development process to conceive, plan and create an app from start to finish.
● Use the AP CSP reference sheet effectively during the exam
● Use a trace table to trace through pseudocode.</td>
</tr>
</table>

Learning Objective 3.3.1 Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. **<u>Big Idea 5 Programming:</u>** Learning Objective 5.1.2 Develop a correct program to solve problems. Learning Objective 5.3.1 Use abstraction to manage complexity in programs. Learning Objective 5.5.1 Employ appropriate mathematical and logical concepts in programming **<u>Big Idea 7 Global Impact:</u>** Learning Objective 7.1.1 Explain how computing innovations affect communication, interaction, and cognition. Learning Objective 7.2.1 Explain how computing has impacted innovations in other fields. Learning Objective 7.3.1 Analyze the beneficial and harmful effects of computing.		• Identify control structures (sequence, selection, repetition) in pseudocode.

Stage 2 – Evidence		
Code	**Evaluative Criteria**	**Assessment Evidence**
M, T	College Board Rubric	PERFORMANCE TASK(S): **CREATE PT 2** is the official CREATE programming performance task to be submitted to the College Board. The CREATE task is one of two required performance tasks by the College Board - a programming one (CREATE) and a written one (EXPLORE). In this programming performance task, students work in pairs to collaboratively develop a mobile app. This includes going through the entire development process of designing, implementing, and debugging a mobile app. Students then document their work by creating a portfolio write-up and share their work through an oral presentation to the class or a recorded video presentation.
M, T	Multiple Choice, C	OTHER EVIDENCE: **Mobile CSP Final Exam** is the last exam for the course which follows the same format at the AP CSP exam. This exam is cumulative and covers Unit 1-7 . It can be used as a practice in review for the AP CSP Exam in May.

Stage 3 – Learning Plan		
Code	***Pre-Assessment*** Think-pair-share regarding standardized tests and strategies students employ when taking them.	
	Summary of Key Learning Events and Instruction	Progress Monitoring
A, T	Discussing the format of the AP CSP exam and reviewing the AP CSP Exam Reference Sheet and AP CSP Pseudocode with Tracing Pseudocode Exercises.	Successful completion of exercises
M	Sample AP CSP Exam Questions and a Mobile CS Principles Quiz app for reviewing and practicing.	Successful completion of questions and monitor time spent on quiz app

Unit 9: Beyond the APCSP Exam

<table>
<tr><th colspan="3">Stage 1 Desired Results</th></tr>
<tr><td rowspan="4">ESTABLISHED GOALS

Computational Thinking Practice 1: Connecting Computing
Identify impacts of computing.
Describe connections between people and computing.
Explain connections between computing concepts.

Computational Thinking Practice 2: Creating Computational Artifacts
Create a computational artifact with a practical, personal, or societal intent.
Select appropriate techniques to develop a computational artifact.
Use appropriate algorithms and information management principles.

Computational Thinking Practice 3: Abstracting
Explain how data, information, or knowledge is represented for computational use.
Explain how abstractions are used in computation or modeling.
Identify abstractions
Describe modeling in a computational context.

Computational Thinking Practice 4: Analyzing Problems and Artifacts</td><td colspan="2">Transfer</td></tr>
<tr><td colspan="2">Students will be able to independently use their learning to…

Apply the problem solving process independently or collaboratively to solve problems.</td></tr>
<tr><td colspan="2">Meaning</td></tr>
<tr><td>UNDERSTANDINGS
Students will understand that…

APCSP Big Idea 1: Creativity.
1.2 Computing enables people to use creative development processes when using computing tools and techniques to create computational artifacts for creative expression of ideas or to solve a problem.

APCSP Big Idea 3: Data and Information
3.1 People use computer programs to process information to gain insight and knowledge.
3.2 Computing facilitates exploration and the discovery of connections in information.
3.3 There are trade-offs when representing information as digital data.

APCSP Big Idea 5: Programming
5.1 Programs can be developed to solve problems (to help people, organizations or society); for creative expression; to satisfy personal curiosity or to create new knowledge.
5.2 People write programs to execute algorithms.</td><td>ESSENTIAL QUESTIONS
Students will keep considering…

How are programs developed to help people, organizations, or society solve problems?

How are programs used for creative expression, to satisfy personal curiosity, or to create new knowledge?</td></tr>
</table>

<table>
<tr>
<td rowspan="3">Evaluate a proposed solution to a problem.
Locate and correct errors
Explain how an artifact functions
Justify appropriateness and correctness of a solution, model, or artifact.

<u>APCSP Learning Objectives:</u>

<u>Big Idea 1 Creativity:</u>

Learning Objective 1.2.1 Create a computational artifact for creative expression.
Learning Objective 1.2.2 Create a computational artifact using computing tools and techniques to solve a problem

<u>Big Idea 3 Data and Information:</u>

Learning Objective 3.1.1 Use computers to process information, find patterns, and test hypotheses about digitally processed information to gain insight and knowledge.
Learning Objective 3.1.2 Collaborate when processing information to gain insight and knowledge

Learning Objective 3.1.3 Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language.

Learning Objective 3.2.1 Extract information from data to discover and explain connections, patterns, or trends.

Learning Objective 3.2.2 Use large data sets to explore and discover information and knowledge.

Learning Objective 3.3.1 Analyze how data</td>
<td>5.3 Programming is facilitated by appropriate abstractions.
5.5 Programming uses mathematical and logical concepts.

APCSP Big Idea 7: Global Impact
7.1 Computing enhances communication, interaction, and cognition.
7.2 Computing enables innovation in nearly every field.
7.3 Computing has a global affect - both beneficial and harmful - on people and society.</td>
<td></td>
</tr>
<tr>
<td colspan="2">Acquisition</td>
</tr>
<tr>
<td>Students will know…

● There are many different pathways to computing careers.
● There are computing jobs in almost all industries, not just technology.
● Computational thinking and problem solving are vital skills that can be applied to many other disciplines, not just computer science.</td>
<td>Students will be skilled at…

● Applying the software development process to conceive, plan and create an app from start to finish.
● Identifying different industries where workers who are skilled at computer science are employed.
● Identifying possible career pathways for themselves that involve computer science.</td>
</tr>
</table>

representation, storage, security, and transmission of data involve computational manipulation of information. **<u>Big Idea 5 Programming:</u>** Learning Objective 5.1.2 Develop a correct program to solve problems. Learning Objective 5.3.1 Use abstraction to manage complexity in programs. Learning Objective 5.5.1 Employ appropriate mathematical and logical concepts in programming **<u>Big Idea 7 Global Impact:</u>** Learning Objective 7.1.1 Explain how computing innovations affect communication, interaction, and cognition. Learning Objective 7.2.1 Explain how computing has impacted innovations in other fields. Learning Objective 7.3.1 Analyze the beneficial and harmful effects of computing.		

Stage 2 – Evidence		
Code	**Evaluative Criteria**	**Assessment Evidence**
		PERFORMANCE TASK(S): Students will complete at least two of the tasks described in the learning events below
		OTHER EVIDENCE:

Stage 3 – Learning Plan		
Code	***Pre-Assessment*** ***The following activities are all optional. Students will be invited to complete whatever activities are appropriate for their level of understanding and expertise.***	
	Summary of Key Learning Events and Instruction	Progress Monitoring
A, T	Teacher directs students to Magic 8 Ball tutorial. App Inventor simulation of the classic Magic-8 Ball game. Introduces the use of a list variable and random selection from the list. A ListPicker is used to implement a simple settings menu that allows the user to select from Speak, Sound, or Silent options for the feedback provided by the app. An if/else algorithm and a global variable are used to implement the setting. Teacher circulates for assistance and enforces the "ask three before me" rule.	App completion checklist.
A, T	Teacher directs students to Persisting Photos Tutorial. This tutorial plus projects lesson shows how to save photos to TinyDb, a simple on-device database. An if/else algorithm is needed to properly initialize the app when initially reading from the Db. A second project uses a simple list to store multiple photos in the Db.	App completion checklist.
M, T	Teacher directs students to complete the Where is North app in pairs using the navigator-driver model. Simple compass app that also reports the device's location. Challenging abstraction exercise: Draw direction markers, N, S, E, W, centered along the edges of the Canvas.	App completion checklist.
M, T	Other optional app-based projects: My Directions. Uses the devices GPS to provide directions from current location to pre-set list of destinations. The Pong Game, Debugging Pong Multiple Choice Quiz App: List of Lists	App completion checklists.

	Hello World Fusion Table App. An optional lesson that uses a Web Viewer to display Fusiontable data No Texting While Busy. The Texting component is used to respond automatically to incoming Text messages. Permits Texting over Wifi (VoIP) by using Google Voice. Research Projects: Learn More About Programming & Career Paths	