PATERSON PUBLIC SCHOOLS
Preparing All Children for College and Career

Department of College and Career Readiness

AP Mobile Computer Science Principles
Curriculum

5.0 Credits

Unit One

AP Mobile Computer Science Principles

Course Description

Mobile CSP is a project-based course based on the AP CS Principles Framework. The course covers the **7 Big Ideas** and **6 Computational Thinking Practices**. During the course, students complete two collaborative programming projects and an individual research and writing project on the impact of a recent, computing innovation that appeals to the student. These projects conform to the College Board's two **performance tasks** on programming and impact. The emerging CS Principles AP course will use these performance tasks, in addition to a written exam, as a primary means for a student to demonstrate what they've learned.

Twenty-eight lessons and projects focus on building socially useful mobile apps with App Inventor for Android. Another 30 lessons focus on computer science topics ranging from algorithms to binary numbers to computer security. Readings from Blown to Bits ask students to reflect on some of the big societal issues that characterize 21st century computing, such as privacy, security, social networking.

AP CS Principles Exam

Students who complete this course will be prepared to take the AP CS Principles Exam. The AP CS Principles Framework is followed in conjunction with the official Mobile CSP Syllabus.

Prerequisites (As described by the College Board)

It is recommended that a student in the AP Computer Science Principles course should have successfully completed a first-year high school algebra course with a strong foundation in basic algebraic concepts dealing with function notation, such as $f(x) = 5x^2$ and problem-solving strategies that require multiple approaches and collaborative efforts. In addition, students should be able to use a Cartesian (x, y) coordinate system to represent points on a plane. It is important that students and their advisers understand that any significant computer science course builds upon a foundation of mathematical reasoning that should be acquired before attempting such a course.

Programming Environment:

App Inventor for Android (ai2.appinventor.mit.edu), a free online software platform, is used in this course to build mobile apps for Android devices.

Online Resources:

The complete curriculum is hosted online and free of charge: https://ram8647.appspot.com/mobileCSP. The course uses many freely available resources that are only available online to ensure that the course material is current and adaptable. Students maintain individual online portfolios of their course work by using Google sites (https://www.google.com/sites/overview.html). Self-check and live coding exercises make use of Quizly (https://github.com/ram8647/quizly), a Web-based live coding platform for App Inventor. Throughout the course, students will also use a number of online articles and videos from sources such as The New York Times (www.nytimes.com), Wikipedia (www.wikipedia.org), CS Bits and Bytes (http://www.nsf.gov/cise/csbytes/), Logic.ly (www.logic.ly), YouTube (www.youtube.com), and CS Unplugged (http://csunplugged.org).

Reference Text:

- App Inventor 2: Create Your Own Android Apps. David Wolber, Hal Abelson, Ellen Spertus, and Liz Looney O'Reilly Media, Inc., 2014
- Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion. Hal Abelson, Ken Ledeen, Harry Lewis. Addison-Wesley, 2010

AP Mobile Computer Science Principles

Pacing Guide		
Unit	**Topic**	**Suggested Timing**
Unit 1	Getting Started: Preview & Setup Introduction to Mobile Apps & Pair Programming Creating Graphics & Images Bit by Bit	approx. 8 weeks
Unit 2	Create Task: Programming Performance Task #1 Exploring Computing: Animation, Simulation, & Modeling Explore Task: Impact of Computing Innovations Performance Task #1	approx. 9 weeks
Unit 3	Algorithms and Procedural Abstractions Using and Analyzing Data & Information Explore Task: Impact of Computing Innovations Performance Task #2	approx. 9 weeks
Unit 4	Communication Through The Internet Create Task: Programming Performance Task #2 Data Project (Optional)	approx. 9 weeks

Educational Technology Standards

8.1.12.A.1, 8.1.12.A.2, 8.1.12.A.3, 8.1.12.B.2, 8.1.12.C.1, 8.1.12.D.1, 8.1.12.D.4, 8.1.12.D.5,8.1.12.E.2, 8.1.12.F.1

- **Technology Operations and Concepts**
  - Create a personal digital portfolio which reflects personal and academic interests, achievements, and career aspirations by using a variety of digital tools and resources.
  - Produce and edit a multi-page digital document for a commercial or professional audience and present it to peers and/or professionals in that related area for review.
  - Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
- **Creativity and Innovation**
  - Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
- **Communication and Collaboration**
  - Develop an innovative solution to a real world problem or issue in collaboration with peers and experts, and present ideas for feedback through social media or in an online community.
- **Digital Citizenship**
  - Demonstrate appropriate application of copyright, fair use and/or Creative Commons to an original work.
  - Research and understand the positive and negative impact of one's digital footprint.
  - Analyze the capabilities and limitations of current and emerging technology resources and assess their potential to address personal, social, lifelong learning, and career needs.
- **Research and Information Literacy**
  - Research and evaluate the impact on society of the unethical use of digital tools and present your research to peers.
- **Critical Thinking, Problem Solving, Decision Making**
  - Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.

Career Ready Practices

Career Ready Practices describe the career-ready skills that all educators in all content areas should seek to develop in their students. They are practices that have been linked to increase college, career, and life success. Career Ready Practices should be taught and reinforced in all career exploration and preparation programs with increasingly higher levels of complexity and expectation as a student advances through a program of study.

CRP1. Act as a responsible and contributing citizen and employee
Career-ready individuals understand the obligations and responsibilities of being a member of a community, and they demonstrate this understanding every day through their interactions with others. They are conscientious of the impacts of their decisions on others and the environment around them. They think about the near-term and long-term consequences of their actions and seek to act in ways that contribute to the betterment of their teams, families, community and workplace. They are reliable and consistent in going beyond the minimum expectation and in participating in activities that serve the greater good.

CRP2. Apply appropriate academic and technical skills.
Career-ready individuals readily access and use the knowledge and skills acquired through experience and education to be more productive. They make connections between abstract concepts with real-world applications, and they make correct insights about when it is appropriate to apply the use of an academic skill in a workplace situation.

CRP4. Communicate clearly and effectively and with reason.
Career-ready individuals communicate thoughts, ideas, and action plans with clarity, whether using written, verbal, and/or visual methods. They communicate in the workplace with clarity and purpose to make maximum use of their own and others' time. They are excellent writers; they master conventions, word choice, and organization, and use effective tone and presentation skills to articulate ideas. They are skilled at interacting with others; they are active listeners and speak clearly and with purpose. Career-ready individuals think about the audience for their communication and prepare accordingly to ensure the desired outcome.

CRP5. Consider the environmental, social and economic impacts of decisions.
Career-ready individuals understand the interrelated nature of their actions and regularly make decisions that positively impact and/or mitigate negative impact on other people, organization, and the environment. They are aware of and utilize new technologies, understandings, procedures, materials, and regulations affecting the nature of their work as it relates to the impact on the social

Career Ready Practices

condition, the environment and the profitability of the organization.

CRP6. Demonstrate creativity and innovation.
Career-ready individuals regularly think of ideas that solve problems in new and different ways, and they contribute those ideas in a useful and productive manner to improve their organization. They can consider unconventional ideas and suggestions as solutions to issues, tasks or problems, and they discern which ideas and suggestions will add greatest value. They seek new methods, practices, and ideas from a variety of sources and seek to apply those ideas to their own workplace. They take action on their ideas and understand how to bring innovation to an organization.

CRP7. Employ valid and reliable research strategies.
Career-ready individuals are discerning in accepting and using new information to make decisions, change practices or inform strategies. They use reliable research process to search for new information. They evaluate the validity of sources when considering the use and adoption of external information or practices in their workplace situation.

CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.
Career-ready individuals readily recognize problems in the workplace, understand the nature of the problem, and devise effective plans to solve the problem. They are aware of problems when they occur and take action quickly to address the problem; they thoughtfully investigate the root cause of the problem prior to introducing solutions. They carefully consider the options to solve the problem. Once a solution is agreed upon, they follow through to ensure the problem is solved, whether through their own actions or the actions of others.

CRP9. Model integrity, ethical leadership and effective management.
Career-ready individuals consistently act in ways that align personal and community-held ideals and principles while employing strategies to positively influence others in the workplace. They have a clear understanding of integrity and act on this understanding in every decision. They use a variety of means to positively impact the directions and actions of a team or organization, and they apply insights into human behavior to change others' action, attitudes and/or beliefs. They recognize the near-term and long-term effects that management's actions and attitudes can have on productivity, morals and organizational culture.

CRP11. Use technology to enhance productivity.

Career Ready Practices
Career-ready individuals find and maximize the productive value of existing and new technology to accomplish workplace tasks and solve workplace problems. They are flexible and adaptive in acquiring new technology. They are proficient with ubiquitous technology applications. They understand the inherent risks-personal and organizational-of technology applications, and they take actions to prevent or mitigate these risks. **CRP12. Work productively in teams while using cultural global competence.** Career-ready individuals positively contribute to every team, whether formal or informal. They apply an awareness of cultural difference to avoid barriers to productive and positive interaction. They find ways to increase the engagement and contribution of all team members. They plan and facilitate effective team meetings.

Differentiated Instruction

Strategies to Accommodate Students Based on Individual Needs

Time/General	Processing	Comprehension	Recall
• Extra time for assigned tasks • Adjust length of assignment • Timeline with due dates for reports and projects • Communication system between home and school • Provide lecture notes/outline	• Extra Response time • Have students verbalize steps • Repeat, clarify or reword directions • Mini-breaks between tasks • Provide a warning for transitions • Reading partners	• Precise step-by-step directions • Short manageable tasks • Brief and concrete directions • Provide immediate feedback • Small group instruction • Emphasize multi-sensory learning	• Teacher-made checklist • Use visual graphic organizers • Reference resources to promote independence • Visual and verbal reminders • Graphic organizers
Assistive Technology	**Tests/Quizzes/Grading**	**Behavior/Attention**	**Organization**
• Computer/whiteboard • Tape recorder • Spell-checker • Audio-taped books	• Extended time • Study guides • Shortened tests • Read directions aloud	• Consistent daily structured routine • Simple and clear classroom rules • Frequent feedback	• Individual daily planner • Display a written agenda • Note-taking assistance • Color code materials

Enrichment

Strategies Used to Accommodate Based on Students Individual Needs:

- Adaption of Material and Requirements
- Evaluate Vocabulary
- Additional Projects
- Independent Student Options
- Projects completed individual or with Partners
- Self-Selection of Research
- Tiered/Multilevel Activities
- Learning Centers
- Individual Response Board
- Independent Book Studies
- Open-ended activities
- Community/Subject expert mentorships

Assessments

Suggested Formative/Summative Classroom Assessments

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

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

Labs: This course will be taught in a computer lab. Students will have access to computers and mobile devices and any other necessary hardware, both during the class and during free periods. Students can work in the lab during their free periods. Internet access will be available to students throughout the course. In each unit, there will be at least three labs designed to practice and/or reinforce key concepts. Some are unplugged and others are completed in an online development environment. Most are completed in App Inventor.

Projects There will be two (2) creative programming projects in which students will use lab time to work both individually and collaboratively (in pairs) to create a socially useful mobile app that they propose (pitch), design, and implement. One of these will be a practice for the College Board's Create Performance Task. The second will be the official College Board Create Performance Task. Twelve (12) hours of class time will be provided for completion of the official Create Performance Task.

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

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

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

Self-check and Live coding exercises: All lessons in this course are accompanied by short, interactive, self-check exercises that consist of multiple choice and fill-in question as well as automatically graded, live-coding, programming exercises (https://github.com/ram8647/quizly). These assessments are considered an essential part of the learning process. These are hosted online and may be done individually or with the class as a whole. Each question or exercise includes detailed feedback and students may repeat the question or exercise until it is correct.

AP Computer Science Principles Exam (AP Exam May 5, 2017) Students who complete this course will be prepared to take the AP CS Principles Exam.

Interdisciplinary Connections

English Language Arts
- Journal writing
- Close reading of industry-related content
- Create a brochure for a specific industry
- Keep a running word wall of industry vocabulary

Social Studies
- Research the history of a given industry/profession
- Research prominent historical individuals in a given industry/profession
- Use historical references to solve problems

World Language
- Translate industry-content
- Create a translated index of industry vocabulary
- Generate a translated list of words and phrases related to information technology

Math
- Compare and contrast use of equations and variables in algebra and programming.
- Program graphics and use the properties of geometric shapes
- Compare the computer graphic coordinate system with the Cartesian coordinate plane in math
- Compare probability and the use of random numbers in computer programming.
- Track and track various data, such as industry's impact on the GDP, career opportunities or among of individuals currently occupying careers

Fine & Performing Arts
- Create a poster recruiting young people to focus their studies on a career in Information Technology

Science
- Research the environmental impact of a given career or industry
- Research latest developments in Information technology
- Investigate applicable-careers in STEM fields

New Jersey Student Learning Standards

8.2 Technology Education, Engineering, Design, and Computational Thinking - Programming

Technology and Society

- 8.2.12.B.3: Analyze ethical and unethical practices around intellectual property rights as influenced by human wants and/or needs.

Design

- 8.2.12.C.1: Explain how open source technologies follow the design process.

Computational Thinking: Programming

- 8.2.12.E.1: Demonstrate an understanding of the problem-solving capacity of computers in our world.
- 8.2.12.E.2: Analyze the relationships between internal and external computer components.
- 8.2.12.E.3: Use a programming language to solve problems or accomplish a task.
- 8.2.12.E.4: Use appropriate terms in conversation.

9.3– Career and Technical Education

Career Cluster: Information Technology (IT)

- 9.3.12.IT.11: Demonstrate knowledge of the hardware components associated with information systems.
- 9.3.12.IT-SUP.9: Employ technical writing and documentation skills in support of an information system.

Pathway: Programming & Software Development (IT-PRG)

- 9.3.12.IT-PRG.4: Demonstrate the effective use of software development tools to develop software applications.
- 9.3.12.IT-PRG.5: Apply an appropriate software development process to design a software application.
- 9.3.12.IT-PRG.6: Program a computer application using the appropriate programming language.
- 9.3.12.IT-PRG.7: Demonstrate software testing procedures to ensure quality products.

Common Career Technical Core (CCTC)
Career Cluster Information Technology

IT.11 – Demonstrate knowledge of the hardware components associated with information systems.

- IT.11.1 - None available at this time.

IT-SUP.9 - Employ technical writing and documentation skills in support of an information system.

- IT-SUP.9.3 - Design technical documentation.

IT-PRG.4 - Demonstrate the effective use of software development tools to develop software applications.

- IT-PRG.4.1 - Employ tools in developing software applications.
- IT-PRG.4.3 - Apply language-specific programming tools/techniques.

IT-PRG.5 - Apply an appropriate software development process to design a software application.

- IT-PRG.5.1 - Describe software development processes and methodology.

IT-PRG.6 - Program a computer application using the appropriate programming language.

- IT-PRG.6.1 - Explain programming language concepts.
- IT-PRG.6.2 - Summarize program development methodology.
- IT-PRG.6.3 - Demonstrate proficiency in developing an application using an appropriate programming language.
- IT-PRG.6.4 - Explain basic software systems implementation.
- IT-PRG.6.6 - Resolve problems with integration.

IT-PRG.7 - Demonstrate software testing procedures to ensure quality products.

- IT-PRG.7.1 - Develop a software test plan.

Common Core State Standards (CCSS)

CCSS - English-Language Arts

Key Ideas and Details:

- CCSS.ELA-LITERACY.RL.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.

Integration of Knowledge and Ideas:

- CCSS.ELA-LITERACY.W.11-12.1 Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.

Production and Distribution of Writing:

- CCSS.ELA-LITERACY.W.11-12.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.

Research to Build and Present Knowledge:

- CCSS.ELA-LITERACY.W.11-12.7 Conduct short as well as more sustained research projects to answer a question (including a self-generated question) or solve a problem; narrow or broaden the inquiry when appropriate; synthesize multiple sources on the subject, demonstrating understanding of the subject under investigation.

Range of Writing:

- CCSS.ELA-LITERACY.W.11-12.10 Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of tasks, purposes, and audiences

Common Core State Standards (CCSS)

CCSS - Mathematics

Reason quantitatively and use units to solve problems:

- CCSS.MATH.CONTENT.HSN-Q.A.2 Define appropriate quantities for the purpose of descriptive modeling.

Create equations that describe numbers or relationships:

- CCSS.MATH.CONTENT.HSA-CED.A.1 Create equations and inequalities in one variable and use them to solve problems.

Analyze functions using different representations:

- CCSS.MATH.CONTENT.HSF-IF.C.7 Graph functions expressed symbolically and show key features of the graph.

Apply geometric concepts in modeling situations:

- CCSS.MATH.CONTENT.HSG-MG.A.1 Use geometric shapes, their measures, and their properties to describe objects

Calculate expected values and use them to solve problems:

- CCSS.MATH.CONTENT.HSS-MD.A.1 Define a random variable for a quantity of interest by assigning a numerical value to each event in a sample space;
- CCSS.MATH.CONTENT.HSS-MD.A.2 Calculate the expected value of a random variable;

Practice Standards - Mathematics

- 1. Make sense of problems and persevere in solving them.
- 2. Reason abstractly and quantitatively.
- 3. Construct viable arguments and critique the reasoning of others.
- 4. Model with mathematics.
- 5. Use appropriate tools strategically.
- 6. Attend to precision.
- 7. Look for and make use of structure.
- 8. Look for and express regularity in repeated reasoning.

Course: AP Mobile CSP **Unit:** I – Getting Started: Preview & Setup Introduction to Mobile Apps & Pair Programming Creating Graphics & Images Bit by Bit **Grade Level:** 9-12	Unit Overview: Unit 1 of the course provides a brief overview of the Mobile CSP curriculum, emphasizing its main theme: learning the principles of computer science while building socially useful mobile apps. The hands-on work focuses on setting up the student's environment, including their programming environment and online portfolios. Students are led through the process of creating a Gmail account, registering on the App Inventor site, and setting up their Google sites portfolio. Their portfolios will be used to display and share all of their written work for the course. Students are provided a brief introduction to blocks-based programming by having them work through a series of increasingly challenging Blockly Maze problems. And they are given a brief introduction to the Blown to Bits book, which is used as a reading resource throughout the course. The unit also provides an introduction to the App Inventor programming platform and the course's first programming project, the I Have a Dream app, a sound board app. Students are introduced to App Inventor's event-driven programming model. Students first work through a guided tutorial that plays an excerpt of a Martin Luther King speech and are then presented with several exercises that challenge them to extend their understanding by solving problems on their own, working in pairs. This is followed later in the unit by several creative mini projects where students are invited to express their own ideas by developing their own computational artifacts. Students are also introduced to several important CS Principles themes and topics. Two lessons focus on hardware and software concepts. The big idea of abstraction is introduced. Students get their first look at binary numbers learning how to count in binary and how to view number systems such as binary, hexadecimal and decimal, as instances of the higher-order abstraction of a positional number system. Finally, the unit extends the student's mobile programming toolkit to several new App Inventor components and introduces a number of new programming concepts, including the concept of a variables, lists and data abstraction. The main app in this unit, The Paint Pot app, a computational version of finger painting, focuses on App Inventor's drawing and painting features and related topics from the CS Principles framework. The app is presented in four parts each of which is followed by a set of creative project exercises and challenges. This unit also introduces two other apps: Magic 8 Ball app, which provides a first introduction to lists, and Map Tour, which demonstrates how to incorporate external data into a mobile app. Unit 3 also extends the student's understanding of binary number system and introduces students to the idea of a bit as the fundamental unit of data. Through a number of hands-on and interactive activities students explore how bits are used to represent images, and how redundant parity bits can be used to detect simple data transmission errors. These lessons are complemented nicely by a Blown to Bits reading that focuses on digital.
New Jersey Student Learning Standards (NJSLS): 8.2.12.B.3, 8.2.12.C.1, 8.2.12.E.1, 8.2.12.E.2, 8.2.12.E.3, 8.2.12.E.4 9.3.12.IT.11, 9.3.12.IT-SUP.9.3, 9.3.12.IT-PRG.4.1, 9.3.12.IT-PRG.4.3, 9.3.12.IT-PRG.5.1, 9.3.12.IT-PRG.6.1, 9.3.12.IT-PRG.6.3, 9.3.12.IT-PRG.6.4, 9.3.12.IT-PRG.6.6, 9.3.12.IT-PRG.7.1	
Common Career Technical Core (CCTC): IT.11, IT-SUP.9.3, IT-PRG.4.1, IT-PRG.4.3, IT-PRG.5.1, IT-PRG.6.1, IT-PRG.6.3, IT-PRG.6.4, IT-PRG.6.6, IT-PRG.7.1	
Common Core State Standards (CCSS): RL.11-12.1; W.11-12.1; W.11-12.4; W.11-12.10; HSN-Q.A.2;	

AP CONTENT STANDARDS: https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf

Big Ideas Standard:
BI 1 – Computing is a creative activity.
BI 2 – Abstraction reduces information and detail to facilitate focus on relevant concepts.
BI 3 – Data and information facilitate the creation of knowledge.
BI 5 – Programming enables problem solving, human expression, and creation of knowledge.
BI 7 – Computing has global Impact

Computational Thinking Practices:
P1 - Study the effects and they learn to draw connections between different computing concepts.
P2 - Creating Computational Artifacts - engage in the creative aspects of computing by designing and developing interesting computational artifacts as well as by applying computing techniques to creatively solve problems.
P3 – Abstracting - use abstraction to develop models and simulations of natural and artificial phenomena, use them to make predictions about the world, and analyze their efficacy and validity.
P4 - Analyzing Problems and Artifacts - design and produce solutions, models, and artifacts, and they evaluate and analyze their own computational work as well as the computational work others have produced.
P5 – Communicating - describe computation and the impact of technology and computation, explain and justify the design and appropriateness of their computational choices, and analyze and describe both computational artifacts and the results or behaviors of such artifacts.
P6 – Collaborating - collaborate on a number of activities, including investigation of questions using data sets and in the production of computational artifacts.

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
ESSENTIAL KNOWLEDGE, SKILLS, AND ENDURING UNDERSTANDINGS: ***A: STUDENTS WILL KNOW THAT:*** • EK 1.1.1A A creative process in the development of a computational artifact can include, but is not limited to, employing nontraditional, non-prescribed techniques; the use of novel combinations of artifacts, tools, and techniques; and the	• BI1 - How can a creative development process affect the creation of computational artifacts? • BI1 - How can computing and the use of computational tools foster creative expression? • BI1 - How can computing extend traditional forms of human expression and experience? • BI4 - How are algorithms implemented and executed on	***STUDENTS WILL BE ABLE TO:*** • LO 1.1.1 Apply a creative development process when creating computational artifacts. [P2] • LO 4.1.1 [P2] – Develop an algorithm for implementation in a program. • LO 1.2.1 Create a computational artifact for creative expression. [P2] • LO 4.1.1 Develop an algorithm	• Welcome to Mobile CSP, • Mazes, Algorithms, and Programs, • Google Account and Portfolio Setup, • App Inventor Setup, • Blown to Bits (BB), • Joining the Forum • Wrap up	Mobile CSP Unit 1 Student Site Mobile CSP Unit 1 Teacher Site Mobile CSP Resources

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
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 4.1.1A Sequencing, selection, and iteration are building blocks of algorithms. • EK 4.1.1B Sequencing is the application of each step of an algorithm in the order in which the statements are given. • EK 4.1.1D Iteration is the repetition of art of an algorithm until a condition is met or for a specified number of times. • EK 4.1.1G Knowledge of standard algorithms can help in constructing new algorithms. • EK 4.1.1I Developing a new algorithm to solve a problem can yield insight into the problem. • 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 4.1.2E Some programming languages are designed for specific domains and are better for expressing algorithms in	computers and computational devices? • BI4 - Why are some languages better than others when used to implement algorithms? • BI 4 - What kinds of problems are easy, what kinds are difficult, and what kinds are impossible to solve algorithmically? • BI4 - How are algorithms evaluated? • BI5 - How are programs developed to help people, organizations, or society solve problems? • BI5 - How are programs used for creative expression, to satisfy personal curiosity, or to create new knowledge? • BI5 - How do computer programs implement algorithms? • BI5 - How does abstraction make the development of computer programs possible? • BI5 - How do people develop and test computer programs? • BI5 - Which mathematical and logical concepts are fundamental to computer programming? • What is the Mobile CS Principles course? • What is graphical blocks-based programming? • Why is it important to study the impact of computing technology? • What is the Mobile CS Principles course? • What is graphical blocks-based programming? • Why is it important to study the impact of computing technology?	for implementation in a program. [P2] • LO 4.1.2 [P5] – Express an algorithm in a language. • LO 4.2.4 [P4] – Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. • LO 5.1.2 [P2] - Develop a correct program to solve problems. • LO 5.1.3 [P6] – Collaborate to develop a program. • LO 5.2.1 [P3] - Explain how programs implement algorithms.		

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
those domains. • 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 5.1.2G Program development includes identifying programmer and user concerns that affect the solution to problems. • EK 5.1.2I A programmer's knowledge and skill affects how a program is developed and how it is used to solve a problem. • EK 5.1.2J A programmer designs, implements, tests, debugs, and maintains programs when solving problems. • 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.				

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
• EK 5.2.1D An understanding of instruction processing and program execution is useful for programming. ***B: STUDENTS WILL UNDERSTAND THAT:*** • EU 1.1 Creative development can be an essential process for creating computational artifacts.EU 4.1 Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages. • EU 1.2 Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem. • EU 4.1 Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages. • EU 4.2 Algorithms can solve many, but not all, computational problems. • EU 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). • EU 5.2 People write programs to execute algorithms.				

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
NJSLS: 8.1.12.A.1, 8.1.12.A.3, 8.2.12.B.3, 8.2.12.E.1, 8.2.12.E.4, 9.3.12.IT.11.1, 9.3.12.IT-PRG.5.1, 8.2.12.E.2, 9.3.12.IT.11.1, 9.3.12.IT.PRG.5.1 **CCTC:** IT.11.1, IT-PRG.5.1 **CCSS:** W.11-12.4, W.11-12.1, W.11-12.4, W.11-12.10, RL.11-12.1, HSN-Q.A.2 ***A: STUDENTS WILL KNOW:*** • 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 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	**ESSENTIAL QUESTIONS THAT WILL FOCUS TEACHING AND LEARNING:** • BI1 - How can a creative development process affect the creation of computational artifacts? • BI1 - How can computing and the use of computational tools foster creative expression? • BI1 - How can computing extend traditional forms of human expression and experience? • BI2 – How are vastly different kinds of data, physical phenomena, and mathematical concepts represented on a computer? • BI2 – How does abstraction help us in writing programs, creating computational artifacts and solving problems? • BI2 – How can computational models and simulations help generate new understanding and knowledge? • BI5 - How are programs developed to help people, organizations, or society solve problems? • BI5 - How are programs used for creative expression, to satisfy personal curiosity, or to create new knowledge? • BI5 - How do computer programs implement algorithms?	***C: STUDENTS WILL BE ABLE TO:*** • LO 1.2.1 Create a computational artifact for creative expression. [P2] • LO 1.2.3 Create a new computational artifact by combining or modifying existing artifacts. [P2] • LO 2.1.1 Describe the variety of abstractions used to represent data. [P3] • LO 2.2.3 Identify multiple levels of abstractions that are used when writing programs. [P3] • LO 4.1.1 Develop an algorithm for implementation in a program. [P2] • LO 5.1.1 Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2] • LO 5.1.2 Develop a correct program to solve problems. [P2] • LO 5.2.1 Explain how programs implement algorithms. [P3] • LO 5.4.1 Evaluate the correctness of a program. [P4] • LO 7.1.1 Explain how computing innovations affect communication, interaction, and cognition. [P4] • LO 7.3.1 Analyze the beneficial and harmful effects of computing. [P4]	• I Have a Dream Tutorial • I Have a Dream Part 2 • Mobile Apps and Mobile Devices • I Have a Dream Projects • What is Abstraction • Blown to Bits: The Digital Explosion • Binary Numbers • Where is North (A compass app) • Hardware and Software Abstractions • Wrap up	Mobile CSP Unit 2 - Introduction to Mobile Apps & Pair Programming Student Site Mobile CSP Unit 2 - Introduction to Mobile Apps & Pair Programming Teacher Site Google Sites https://sites.google.com/ App Inventor 2 http://ai2.appinventor.mit.edu/

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
converted from any base to any other base. • 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 (for EK 2.2.3A): 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. • EK 2.2.3C Code in a programming language is often translated into code in another (lower level) language to be executed on a computer. • 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 (for EK 2.2.3F): Memorization of specific gate visual representations is beyond the	• BI5 - How does abstraction make the development of computer programs possible? • BI5 - How do people develop and test computer programs? • BI5 - Which mathematical and logical concepts are fundamental to computer programming? • BI7 - How does computing enhance human communication, interaction, and cognition? • BI7 - How does computing enable innovation? • BI7 - What are some potential beneficial and harmful effects of computing? GUIDING QUESTIONS: • How does one use App Inventor and event-driven programming to build a mobile app? • What are the various hardware and software abstractions that make up a modern digital computer? • What is the binary number system that underlies all digital representation?			

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
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 4.1.1C Selection uses a Boolean condition to determine which of two parts of an algorithm is used. • 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.				

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
• EK 5.4.1M The functionality of a program is often described by how a user interacts with it. • 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. ***B: STUDENTS WILL UNDERSTAND THAT:*** • EU 1.2 Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem. • EU 2.1 A variety of abstractions built upon binary sequences can be used to represent all digital data. • EU 2.2 Multiple levels of abstraction are used to write programs or to create other computational artifacts. • EU 5.1 Programs can be developed for creative				

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
expression, to satisfy personal curiosity, to create new knowledge, or to solve problems (to help people, organizations, or society). • EU 5.2 People write programs to execute algorithms. • EU 5.4 Programs are developed, maintained, and used by people for different purposes. • EU 7.1 Computing enhances communication, interaction, and cognition. • EU 7.3 Computing has a global affect — both beneficial and harmful — on people and society.				
NJSLS: 8.2.12.E.1, 8.2.12.E.3, 8.2.12.E.4, 9.3.12.IT.11.1, 9.3.12.IT-PRG.4.1, 9.3.12.IT-PRG.4.3, 9.3.12.IT-PRG.5.1, 9.3.12.IT-PRG.6.1, 9.3.12.IT-PRG.6.3, 9.3.12.IT-PRG.6.4, 9.3.12.IT-PRG.6.6, 9.3.12.IT-SUP.9.3, 9.3.12.IT-PRG.4.3, 9.3.12.IT-PRG.7.1 **CCTC:** IT.11.1, IT-PRG.4.1, IT-PRG.4.3, IT-PRG.5.1, IT-PRG.6.1, IT-PRG.6.3, IT-PRG.6.4, IT-PRG.6.6, IT-PRG.7.1, IT.11.1, IT-SUP.9.3, **CCSS:** W.11-12.4, W.11-12.10 ***A: STUDENTS WILL KNOW:*** • EK 1.2.2A Computing tools and techniques can enhance the process of finding a solution to a	• BI1 - How can a creative development process affect the creation of computational artifacts? • BI1 - How can computing and the use of computational tools foster creative expression? • BI1 - How can computing extend traditional forms of human expression and experience? • BI2 – How are vastly different kinds of data,	***C: STUDENTS WILL BE ABLE TO:*** • LO 1.2.2 Create a computational artifact using computing tools and techniques to solve a problem. [P2] • LO 1.2.4 Collaborate in the creation of computational artifacts. [P6] • LO 2.1.1 Describe the variety of abstractions used to represent data. [P3]	• Paint Pot 1 (A finger painting app), • Paint Pot 1 Projects, • Representing Images, • Blown to Bits: Electronic Documents, • Paint Pot 2 (An introduction to variables), • Paint Pot 2 Projects, • Error Detection, • Magic 8-Ball Tutorial and Projects, • Parity Error Detection, • Map Tour Tutorial and Projects,	Mobile CSP Unit 3 - Creating Graphics & Images Bit by Bit Mobile CSP Unit 3 – Creating Graphics & Images Bit by Bit Teacher Site Google Sites https://sites.google.com/

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
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 developing computational artifacts. • EK 1.2.4F A collaboratively created computational artifact can reflect personal expressions of ideas. • 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 (for EK 2.1.1E): Two's complement conversions are beyond the scope of	physical phenomena, and mathematical concepts represented on a computer? • BI2 – How does abstraction help us in writing programs, creating computational artifacts and solving problems? • BI2 – How can computational models and simulations help generate new understanding and knowledge? • BI3 – How can computation be employed to help people process data and information to gain insight and knowledge? • BI3 – How can computation be employed to facilitate exploration and discovery when working with data? • BI3 – What considerations and trade-offs arise in the computational manipulation of data? • BI3 – What opportunities do large data sets provide	• LO 2.1.2 Explain how binary sequences are used to represent digital data. [P5] • LO 2.2.1 Develop an abstraction when writing a program or creating other computational artifacts. [P2] • LO 2.3.1 Use models and simulations to represent phenomena. [P3] • LO 3.1.2 Collaborate when processing information to gain insight and knowledge. [P6] • LO 3.2.1 Extract information from data to discover and explain connections, patterns, or trends. [P1] • LO 3.3.1 Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. [P4] • LO 5.1.1 Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2]	• Wrap up	App Inventor 2 http://ai2.appinventor.mit.edu/ Chapter Three of Blown to Bits

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
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 (for EK 2.1.2B): 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. • EK 2.2.1A The process of developing an abstraction involves removing detail and generalizing functionality. • 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 3.1.2B Collaboration facilitates solving computational problems by applying multiple perspectives, experiences, and skill sets.	for solving problems and creating knowledge? • BI5 - How are programs developed to help people, organizations, or society solve problems? • BI5 - How are programs used for creative expression, to satisfy personal curiosity, or to create new knowledge? • BI5 - How do computer programs implement algorithms? • BI5 - How does abstraction make the development of computer programs possible? • BI5 - How do people develop and test computer programs? • BI5 - Which mathematical and logical concepts are fundamental to computer programming? • BI7 - How does computing enhance human communication, interaction, and cognition?	• LO 5.1.2 [P2] - Develop a correct program to solve problems. • LO 5.1.3 [P6] – Collaborate to develop a program. • LO 5.3.1 Use abstraction to manage complexity in programs. [P3] • LO 5.4.1 Evaluate the correctness of a program. [P4] • LO 5.5.1 Employ appropriate mathematical and logical concepts in programming. [P1] • LO 7.1.1 Explain how computing innovations affect communication, interaction, and cognition. [P4]		

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
• EK 3.1.2E Collaborating face-to-face and using online collaborative tools can facilitate processing information to gain insight and knowledge. • 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 3.3.1C 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 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 5.1.3A Collaboration can	• BI7 - How does computing enable innovation? • BI7 - What are some potential beneficial and harmful effects of computing? GUIDING QUESTIONS: • How can binary numbers be used to represent all digital data? • How can algorithms be used to compress data? • How do variables of both simple and structured data, such as, lists, enable us manage the complexity of a programming?			

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
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 5.3.1L Using lists and procedures as abstractions in programming can result in programs that are easier to develop and maintain. • 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 5.4.1M The functionality of a program is often described by how a user interacts with it. • EK 5.5.1A Numbers and numerical concepts are fundamental to programming.				

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
• EK 5.5.1D Mathematical expressions using arithmetic operators are part of most programming languages. • EK 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. • EK 7.1.1I Global Positioning System (GPS) and related technologies have changed how humans travel, navigate, and find information related to geolocation. ***B: STUDENTS WILL UNDERSTAND THAT:*** • EU 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 problems. • EU 2.1 A variety of abstractions built upon binary sequences can be used to represent all digital data. • EU 2.2 Multiple levels of abstraction are used to write programs or to create other computational artifacts. • EU 2.3 Models and simulations use abstraction to generate new understanding and knowledge. • EU 3.1 People use computer programs to process information to gain insight and knowledge. • EU 3.3 There are trade-offs when				

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
representing information as digital data. • EU 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. • EU 5.2 People write programs to execute algorithms. • EU 5.3 Programming is facilitated by appropriate abstractions. • EU 5.4 Programs are developed, maintained, and used by people for different purposes. • EU 5.5 Programming uses mathematical and logical concepts. • EU 7.1 Computing enhances communication, interaction, and cognition.				

Unit 1 Vocabulary	
mobile computing algorithm program control structures emulator interface binary abstraction event-driven programming special purpose computers general purpose computers Pixel Parity encoding image compression Internet World Wide Web (WWW)	**Blown to Bits Chapter 3 Vocabulary** algorithm analog ASCII cloud computing cryptography digital digital signal processing download lossless compression lossy compression megabyte megapixel modeling OCR pixel raster render spam steganography upload

Suggested Unit Projects *Choose At Least One*	
Sound Board Project https://ram8647.appspot.com/mobileCSP/unit?unit=1&lesson=47 **Paint Pot Project** https://ram8647.appspot.com/mobileCSP/unit?unit=22&lesson=151 **Magic 8 Ball Project** https://ram8647.appspot.com/mobileCSP/unit?unit=22&lesson=78	**I Have a Dream Projects** The I Have a Dream Projects lesson is the third and culmination of a series of three related lessons: students are invited to express their own ideas and implement their own enhancements and extensions to the app we've been studying. In the first lesson students follow an instructor-led tutorial on how to build a basic sound board app (I Have a Dream). The instructor introduces basic App Inventor programming concepts, including the event-driven programming model that is used throughout the course. In the second lesson, students are given a set of small but increasingly challenging exercises and encouraged to work collaboratively to figure out the solutions on their own. In this culminating lesson, students design and implement enhancements and extensions to the app, including, possibly, creating an entirely new sound board app based on their own ideas and interests. These activities build toward EU 1.1, EU 1.2, EU 1.3, EU 5.1 and EU 5.4 by focusing on creativity, abstraction, and programming concepts. LOs 1.1.1 [P2], 1.2.1 [P2], 1.2.1 [P2], 1.2.3 [P2], 1.2.4 [P6], 1.3.1 [P2], 5.1.1 [P2], 5.4.1 [P4]

Suggested Structured Learning Experiences	
TEALS Student Field Trip to Microsoft https://www.tealsk12.org/events/ Labs: Mazes, Algorithms, and Programs (Blockly), App Inventor Setup (App Inventor), I Have a Dream Tutorial (App Inventor), I Have a Dream Part 1 (App Inventor), I Have a Dream Projects (App Inventor), Where is North (Compass App using App Inventor), Paint Pot 1 (App Inventor), Paint Pot 1 Projects (App Inventor), Paint Pot 2 (App Inventor), Paint Pot 2 Projects (App Inventor), Magic 8-Ball (Using App Inventor), Map Tour (App Inventor and Google Maps Activity Starter)	**InfoAge Science History Museum** https://www.infoage.org