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 Two

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 $f(x) = 5\ x^2$ 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

PATERSON PUBLIC SCHOOLS
Preparing All Children for College and Career

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 Computer Science Principles **Unit: 2** – Create Task Practice: Programming Performance Task #1 Exploring Computing: Animation, Simulation, & Modeling Explore Task: Impact of Computing Innovations Performance Task #1 **Grade Level:** 9-12	Unit Overview: In this unit students work collaboratively with a partner (pair programming) to create a socially useful, interactive, mobile app. The app must in some way include drawing, graphics, and programming constructs based on skills learned in prior lessons. Students are taught how to brainstorm their ideas and develop wireframes with storyboards to express those ideas. Students are asked to give a 2-minute elevator pitch of their app idea and receive feedback from the instructor and their classmates. In class time is given to develop, test, and debug their app. The instructor answers any questions and provides feedback along the way. While working on their app, students are shown how to and asked to maintain a portfolio write up of their work making note of their progress and any challenges they may have faced, as well as, screenshots of blocks of code with written explanations of the how the code works. Students are shown how to record a video of their app. The project ends with an in class presentation and app demo by each pair of students. The unit also focuses on animation, simulation and modeling. The Android Mash app introduces the idea of computer simulation with a computational version of the traditional Whack-a-Mole game. The Coin Flip app, which extends over several lessons, introduces the concept of modeling. The activities in Unit 4 build toward EU 2.3 as students learn that models use abstractions, such as a pseudo random number generator (PRNG), to represent real word situations, in this case, the flipping of a coin; EU 3.3 as students learn how PRNG algorithms are used to model randomness inside a computer, such as with the Coin Flip app; EU 7.1 as students extend the app model to represent different types of coins, including a biased coin and a three-sided coin. This is followed by an experimental lesson where an app that repeatedly "flips" a coin is used to assess the quality of App Inventor's PRNG; EU 7.3 as students learn how one's privacy is impacted by developing technology and computing innovations; and EU 7.4 as students learn the economic, social and cultural effects of computing innovations, such as real world models of the weather and the solar system. Finally, the unit involves discussing, as a class, a computing innovation that has had considerable impact on the social, economic, or cultural areas of our lives, such as phone monitoring software. Students work collaboratively in small groups to research the computing innovation and find reliable sources using sites such as the ACM Digital Library. Students are also asked to cite their sources and are instructed about plagiarism. The instructor assigns each group member a prompt taken from the official Explore Performance Task to answer about the innovation. Each group member answers the prompts in a single Google document that is shared among the group. The group then works together to edit the entire document discussing changes that need to be made. When the document is completed (i.e. all prompts are answered and all sources are cited), each student is asked prepare their own original digital artifact (e.g. music, image, video, infographic, presentation, program, web page) to express the effects the chosen innovation. Students are asked to share their artifact with their group members. After completing this activity, the students are asked to reflect on the experience and to brainstorm at least three computing innovations they might want to research for the official Explore Performance Task.
New Jersey Student Learning Standards (NJSLS): 8.2.12.B.3, 8.2.12.E.1, 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.11-12.7; W.11-12.10; HSN-Q.A.2; HSS-MD.A.1; HSS-MD.A.2;

LINK TO 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 4 – Algorithms are used to develop and express solutions to computational problems.
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
NJSLS: 8.2.12.E.1, 8.2.12.E.3, .2.12.E.4, 9.3.12.IT.11.1, 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	• BI1 - How can a creative development process affect the creation of computational artifacts? • BI1 - How can computing and the use of	***C: STUDENTS WILL BE ABLE TO:*** LO 1.1.1 Apply a creative development process when creating computational artifacts. [P2]	Students work collaboratively with a partner (pair programming) to create a socially useful,	Create: Programming Performance Task #1 Create: Programming Performance Task #1 -

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
CCTC: IT.11.1, 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 **CCSS:** W.11-12.1, W.11-12.4, W.11-12.7, W.11-12.10, HSN-Q.A.2, HSS-MD.A.1, HSS-MD.A.2 ***A: STUDENTS WILL KNOW:*** 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 exploration of personal curiosities. EK 1.1.1B Creating computational artifacts employs an iterative and often exploratory process to translate ideas into tangible form. 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 1.2.1D A creatively developed computational artifact can be created by using nontraditional, nonprescribed computing techniques. EK 1.2.1E Creative expressions in a computational artifact can reflect personal expressions of ideas or interests. EK 1.2.2B A creative development	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? • BI4 - How are algorithms implemented and executed on 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	LO 1.2.1 Create a computational artifact for creative expression. [P2] LO 1.2.2 Create a computational artifact using computing tools and techniques to solve a problem. [P2] LO 1.2.3 Create a new computational artifact by combining or modifying existing artifacts. [P2] LO 1.2.4 Collaborate in the creation of computational artifacts. [P6] LO 1.2.5 Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] LO 2.2.1 Develop an abstraction when writing a program or creating other computational artifacts. [P2] LO 2.2.2 Use multiple levels of abstraction to write programs. [P3] LO 4.1.1 Develop an algorithm for implementation in a program. [P2] LO 4.1.2 Express an algorithm in a language. [P5] LO 5.1.1 Develop a program for creative expression, to satisfy personal curiosity, or to	interactive, mobile app. The app must in some way include drawing, graphics, and programming constructs based on skills learned in prior lessons. Students are taught how to brainstorm their ideas and develop wireframes with storyboards to express those ideas. Students are asked to give a 1-2 minute elevator pitch of their app idea and receive feedback from the instructor and their classmates. In class time is given to develop, test, and debug their app. The instructor answers any questions and provides feedback along the way. While working on their app, students are shown how to and asked to maintain a portfolio write	INSTRUCTOR MATERIALS Creative Commons Licenses Search Creative Commons Media Google Image Search Exam 1 - Midterm

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
process for creating computational artifacts can be used to solve problems when traditional or prescribed computing techniques are not effective. EK 1.2.3A Creating computational artifacts can be done by combining and modifying existing artifacts or by creating new artifacts. EK 1.2.3B Computation facilitates the creation and modification of computational artifacts with enhanced detail and precision. 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 1.2.4C 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 2.2.1A 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 2.2.1C An abstraction generalizes functionality with input parameters that allow software reuse.	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 kind of app would you and your partner like to create? • What programming language will you use to create this app? • What do you want your	create new knowledge. [P2] LO 5.1.2 Develop a correct program to solve problems. [P2] LO 5.1.3 Collaborate to develop a program. [P6] LO 5.2.1 Explain how programs implement algorithms. [P3] 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]	up of their work making note of their progress and any challenges they may have faced, as well as, screenshots of blocks of code with written explanations of the how the code works. Students are shown how to record a video of their app. The project ends with an in class presentation and app demo by each pair of students	

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
EXCLUSION STATEMENT (for EK 2.2.1C): 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 2.2.2B 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 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 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 4.1.1G 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.	app to do? • What is the purpose/function of the app you will create?			

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
EK 4.1.2D Different languages are better suited for expressing different algorithms. EK 5.1.1A Programs are developed and used in a variety of ways by a wide range of people depending on the goals of the programmer. 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 5.1.1C Programs developed for creative expression, to satisfy personal curiosity, or to create new knowledge may be developed with different standards or methods than programs developed for widespread distribution. EK 5.1.1D Additional desired outcomes may be realized independently of the original purpose of the program. EK 5.1.1E A computer program or the results of running a program may be rapidly shared with a large number of users and can have widespread impact on individuals, organizations, and society. 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 5.1.2C Incrementally adding tested program segments to correct working programs helps create large correct programs. EK 5.1.2D Program documentation helps programmers develop and				

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
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 5.1.2G Program development includes identifying programmer and user concerns that affect the solution to problems. EK 5.1.2J A programmer designs, implements, tests, debugs, and maintains programs when solving problems. 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 5.2.1A Algorithms are implemented using program instructions that are processed during				

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
program execution. EK 5.2.1B Program instructions are executed sequentially. EK 5.2.1C Program instructions may involve variables that are initialized and updated, read, and written. EK 5.2.1D An understanding of instruction processing and program execution is useful for programming. 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 5.4.1A Program style can affect the determination of program correctness. EK 5.4.1C Meaningful names for variables and procedures help people better understand programs. EK 5.4.1D Longer code blocks are harder 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 5.4.1F Knowledge of what a program is supposed to do is required in order to find most program errors. 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 because of storage limitations. EXCLUSION STATEMENT (for EK				

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
5.5.1B): 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 (for EK 5.5.1C): Specific sets of values that cannot be exactly represented by floating point numbers are beyond the scope of this course and the AP Exam. EK 5.5.1D Mathematical expressions using arithmetic operators are part of most programming languages. EK 5.5.1F Compound expressions using and, or, and not are part of most programming languages. ***B: STUDENTS WILL UNDERSTAND THAT:*** EU 1.1 Creative development can be an essential process for creating computational artifacts. EU 1.2 Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem. EU 2.2 Multiple levels of abstraction are used to write programs or create other computational artifacts. EU 4.1 Algorithms are precise sequences of instructions for processes that can be executed by a computer and are				

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
implemented using programming languages. 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. 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.				
A: STUDENTS WILL KNOW: EK 1.2.1B Creating computational artifacts requires understanding of and use of software tools and services. EK 1.2.1E Creative expressions in a computational artifact can	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	***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]	• Android Mash Tutorial, • Android Mash Projects • Coin Flip Simulation, • Coin Flip Experiment, • Pseudo Random Number Generators (PRNGs),	**Mobile CSP Unit 4:** ***Animation, Simulation, and Modeling: Exploring the Impact of Computing (Creativity, Abstraction, Data and Information, Algorithms, Programming, & Impact)***

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
reflect personal expressions of ideas or interests. EK 1.2.3C Combining or modifying existing artifacts can show personal expression 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 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 2.3.1A 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	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? 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	LO 1.3.1 Use computing tools and techniques for creative expression. [P2] 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 2.3.2 Use models and simulations to formulate, refine, and test hypotheses. [P3] LO 3.3.1 Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. [P4] LO 4.1.1 Develop an algorithm for implementation in a program. [P2] LO 4.1.2 Express an algorithm in a language. [P5] LO 5.1.2 Develop a correct program to solve problems. [P2] LO 5.3.1 Use abstraction to manage complexity in programs. [P3] LO 5.5.1 Employ appropriate mathematical and logical concepts in programming. [P1]	• Coin Flip Simulation Projects, • Real World Models, • Blown to Bits: Privacy, • Wrap up	Teacher Site Unit 4 – Exploring Computing: Animation, Simulation, & Modeling Student Site Unit 4 - Exploring Computing: Animation, Simulation, & Modeling Chapter 2 of Blown to Bits double entry journal sheet.

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
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 allow hypotheses to be tested without the constraints of the real world. EK 2.3.2F Simulations can facilitate extensive and rapid testing of models.	large data sets provide for solving problems and creating knowledge? BI4 - How are algorithms implemented and executed on 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	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] LO 7.4.1 Explain the connections between computing and economic, social, and cultural contexts. [P1]		

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
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. 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. EK 4.1.1B Sequencing is the application of each step of an algorithm in the order in which the statements are given. EK 4.1.1C Selection uses a Boolean condition to determine which of two parts of an algorithm is used. EK 4.1.2C Algorithms described in programming languages can be executed on a computer. EK 5.1.2C Incrementally adding tested program	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 do computers use simulation and modeling to represent real world phenomena? Why is randomness important and how is it modeled inside a computer? In what ways does simulation and modeling extend our knowledge and benefit			

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
segments to correct working programs helps create large correct 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 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	society?			

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
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 7.4.1A 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. ***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.				

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
EU 1.3 Computing can extend traditional forms of human expression and experience. EU 2.2 Multiple levels of abstraction are used to write programs or create other computational artifacts. EU 2.3 Models and simulations use abstraction to generate new understanding and knowledge. EU 3.3 There are trade-offs when representing information as digital data. 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 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 5.3 Programming is facilitated by appropriate abstractions. EU 5.5 Programming uses mathematical and logical concepts. EU 7.1 Computing enhances				

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
communication, interaction, and cognition. EU 7.3 Computing has a global affect — both beneficial and harmful — on people and society. EU 7.4 Computing innovations influence and are influenced by the economic, social, and cultural contexts in which they are designed and used.				
ESSENTIAL KNOWLEDGE, SKILLS, AND ENDURING UNDERSTANDINGS: ***A: STUDENTS WILL KNOW:*** EU 1.1 Creative development can be an essential process for creating computational artifacts. EU 1.2 Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem. EU 3.1 People use computer programs to process information to gain insight and knowledge. EU 3.3 There are trade-offs when representing information as digital data. 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. EU 7.4 Computing innovations	**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? BI3 – How can computation be employed to help people process data and information to gain insight and	***C: STUDENTS WILL BE ABLE TO:*** LO 1.1.1 Apply a creative development process when creating computational artifacts. [P2] LO 1.2.1 Create a computational artifact for creative expression. [P2] LO 1.2.2 Create a computational artifact using computing tools and techniques to solve a problem. [P2] LO 1.2.3 Create a new computational artifact by combining or modifying existing artifacts. [P2] LO 1.2.5 Analyze the correctness, usability, functionality, and suitability of	Explore: Impact of a Computing Innovation Performance Task #1 (Creativity, Impact) Impact of a computing innovation This activity involves discussing, as a class, a computing innovation that has had considerable impact on the social, economic, or cultural areas of our lives, such as phone monitoring software. Students work collaboratively in small groups to research the computing innovation and find reliable sources using	Explore: Impact of Computing Innovations Performance Task #1 - INSTRUCTOR MATERIALS Explore: Impact of Computing Innovations Performance Task #1 Explore Performance Task AP Fall 2015 Version

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
influence and are influenced by the economic, social, and cultural contexts in which they are designed and used. EU 7.5 - An investigative process is aided by effective organization and selection of resources. Appropriate technologies and tools facilitate the accessing of information and enable the ability to evaluate the credibility of sources. ***B: STUDENTS WILL UNDERSTAND 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 exploration of personal curiosities. EK 1.1.1B Creating computational artifacts employs an iterative and often exploratory process to translate ideas into tangible form. 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 1.2.1D A creatively developed computational artifact can be created by using nontraditional, non-prescribed computing techniques. EK 1.2.1E Creative expressions in a computational artifact can reflect personal expressions of ideas or interests.	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 for solving problems and creating knowledge? 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: What computational research do you want to do? How does this research pertain to computer science?	computational artifacts. [P4] LO 3.1.3 Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language. [P4] LO 3.3.1 Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. [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] LO 7.4.1 Explain the connections between computing and economic, social, and cultural contexts. [P1] LO 7.5.1 Access, manage, and attribute information using effective strategies. [P1] LO 7.5.2 Evaluate online and print sources for appropriateness and credibility. [P1]	sites such as the ACM Digital Library. Students are also asked to cite their sources and are instructed about plagiarism. The instructor assigns each group member a prompt taken from the official Explore Performance Task to answer about the innovation. Each group member answers the prompts in a single Google document that is shared among the group. The group then works together to edit the entire document discussing changes that need to be made. When the document is completed (i.e. all prompts are answered and all sources are cited), each student is asked prepare their own original digital artifact (e.g. music, image, video, infographic, presentation, program, web page) to express the effects the chosen innovation. Students are asked to share their artifact with their	

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
EK 1.2.2B A creative development process for creating computational artifacts can be used to solve problems when traditional or prescribed computing techniques are not effective. EK 1.2.3A Creating computational artifacts can be done by combining and modifying existing artifacts or by creating new artifacts. EK 1.2.5A The context in which an artifact is used determines the correctness, usability, functionality, and suitability of the artifact. EK 1.2.5B A computational artifact may have weaknesses, mistakes, or errors depending on the type of artifact. EK 3.1.3A Visualization tools and software can communicate information about data. EK 3.1.3B Tables, diagrams, and textual displays can be used in communicating insight and knowledge gained from data. EK 3.1.3C Summaries of data analyzed computationally can be effective in communicating insight and knowledge gained from digitally represented information. EK 3.1.3D Transforming information can be effective in communicating knowledge gained from data. EK 3.1.3E Interactivity with data is an aspect of communicating. EK 3.3.1A Digital data representations involve trade-offs related to storage, security, and privacy concerns. EK 3.3.1B Security concerns engender trade-offs in storing and transmitting information. EK 3.3.1E Lossy data compression can significantly reduce the number of	What aspect of computer science is the most interesting to you?		group members. After completing this activity, the students are asked to reflect on the experience and to brainstorm at least three computing innovations they might want to research for the official Explore Performance Task	

Student Learning Objectives (SLOs)	Essential Questions	Skills & Indicators	Sample Activities	Resources
bits stored or transmitted at the cost of being able to reconstruct only an approximation of the original data. EK 3.3.1F Security and privacy concerns arise with data containing personal information. EK 3.3.1G Data is stored in many formats depending on its characteristics (e.g., size and intended use). EK 3.3.1H The choice of storage media affects both the methods and costs of manipulating the data it contains. EK 3.3.1I Reading data and updating data have different storage requirements. 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 7.3.1J Technology enables the collection, use, and exploitation of information about, by, and for individuals, groups, and institutions. 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 7.4.1A 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.				

Unit 2 Vocabulary	
Random Numbers pseudo random number generators Modeling Simulation Canvas Source code abstraction Parity	**Blown to Bits Chapter 2 Vocabulary** ad hoc database data aggregation data mining data repository data sources digital detritus dossier EDR encode encryption IP address metadata query RFID

Suggested Unit Projects

Choose At Least One

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

Explore: Impact of a Computing Innovation Performance Task #1 (Creativity, Impact)
https://ram8647.appspot.com/mobileCSP/assessment?name=123

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

Create an animation.
See Lego Movie Maker App: Animation Tips and Tricks.
https://www.youtube.com/watch?v=mfg9j7ppoaw
https://ram8647.appspot.com/mobileCSP/unit?unit=23&lesson=73

Suggested Structured Learning Experiences

Women attend the annual New Jersey TeenTech conference
http://www.aauw.org/resource/teentech-program-in-a-box/

Museum of the Moving Image
https://www.movingimage.us