BERKELEY HEIGHTS PUBLIC SCHOOLS
BERKELEY HEIGHTS, NEW JERSEY

COLUMBIA MIDDLE SCHOOL
COMPUTER DEPARTMENT

APP DESIGN
Grade 7-8

Curriculum Guide

Date: September 2018
Updated: December 2019

Dr. Melissa Varley, Superintendent
Mr. Scott McKinney, Assistant Superintendent
Mr. James Finley, District Supervisor

Developed by: Michael Riley

This curriculum may be modified through varying techniques, strategies, and materials, as per an individual student's Individualized Educational Plan (IEP).

Approved by the Berkeley Heights Board of Education
at the regular meeting held on 12/5/19 .

VISION STATEMENT

STEM is the integrated approach to education in the areas of Science, Technology, Engineering, and Mathematics. Instruction is student centered and driven by an iterative design process, exploratory learning, problem-solving, and engagement in authentic contexts.

Through the process of engaging in authentic, hands-on, open-ended design challenges, students will become familiar with the steps and processes associated with successful problem solving in the context of the engineering design process. Students will gain proficiency in the application of relevant Math, Science, and Technology concepts while expanding their comprehension and understanding of the human-designed world, the nature of technology and engineered systems, and the skills, knowledge, and attitudes necessary to become well-rounded and successful twenty-first century problem solvers and innovators.

MISSION STATEMENT

App Design is a one quarter cycle class designed for 8th grade students at Columbia Middle school that addresses 21st century skills, career ready practices, and technology standards. Students are introduced to the design process and given the opportunity to utilize these as they work through the app design process. Throughout the course, students will be engaged in an authentic problem-based learning environment working as a team. Students will learn to utilize feedback from peers to revise their designs and develop better solutions. Throughout the course students gain basic programming skills using code.org's CS Discoveries framework. They learn to represent computing processes both on and offline and and consider the impact of technology in the world. Additionally, this course lays the foundational knowledge needed to solve real world problems across different areas of study and outside the classroom.

COURSE PROFICIENCIES

COURSE OBJECTIVES

Having a strategy for approaching problems can help you develop new insights and come up with new and better solutions. This process is generally useful for solving all kinds of problems.

The Problem Solving Process

Define
- What problem are you trying to solve?
- What are your constraints?
- What does success look like?

Prepare
- Brainstorm / research possible solutions
- Compare pros and cons
- Make a plan

Try
- Put your plan into action

Reflect
- How do your results compare to the goals you set while defining the problem?
- What can you learn from this or do better next time?
- What new problems have you discovered?

Similarly, the design process is a loop because although the steps are listed in sequential order, you will likely return to previous steps multiple times throughout a project. It is often necessary to revisit stages or steps in order to improve that aspect of a project.

In the design process,

- Ask (What are we trying to solve?, What are the constraints?, What are the requirements?, What questions do you have about the challenge?)
- Imagine (What are the possible solutions?, Brainstorm ideas, list materials needed, explain the ideas, create a sketch for you ideas)
- Plan (Choose a final solution, sketch a final solution, decide the steps you will take to create your solution, create a technical drawing to explain your design)
- Create (follow your plan and create your design, what changes did you make while creating your design? Why?, Self reflection)
- Improve (What worked well? What could have gone better? What improvements could you make to allow your design to be more successful? Why is the redesign better than the original design?)

Career Ready Practices

CRP1. Act as a responsible and contributing citizen and employee.
CRP2. Apply appropriate academic and technical skills.
CRP4. Communicate clearly and effectively and with reason.

CRP5. Consider the environmental, social and economic impacts of decisions.
CRP6. Demonstrate creativity and innovation.
CRP7. Employ valid and reliable research strategies.
CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.
CRP9. Model integrity, ethical leadership and effective management.
CRP11. Use technology to enhance productivity.
CRP12. Work productively in teams while using cultural global competence.

STUDENT PROFICIENCIES

Students will understand:

- Compare and refine multiple algorithms for the same task and determine which is the most appropriate.
- Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
- Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation and review stages of program development.
- Describe how internal and external parts of computing devices function to form a system.
- Model how computer hardware and software work together as a system to accomplish tasks.
- Use flowcharts and/or pseudocode to address complex problems as algorithms.
- Systematically test and refine programs using a range of test cases
- Compare trade offs associated with computing technologies that affect people's everyday activities and career options.
- Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
- Design projects that combine hardware and software components to collect and exchange data.
- Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
- Discuss issues of bias and accessibility in the design of existing technologies.
- Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
- Document programs in order to make them easier to follow, test, and debug.
- Create clearly named variables that represent different data types and perform operations on their values.
- Incorporate existing code, media, and libraries into original programs, and give attribution.
- Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

METHODS OF EVALUATION

1. Teacher observation/questioning/monitoring
2. Project Notebooks/Journals
3. Team evaluation rubrics
4. Self and peer evaluation
5. Performance tasks/assessments
6. Reports and presentations
7. Student created designs and models
8. Final presentations

MODIFICATIONS & ACCOMMODATIONS

Modifications and Accommodations for Special Education students, students with 504s, English Language Learners and Gifted and Talented students may include but are not limited to the following:

Special Education

- Individualized Education Plans (IEPs)
- Exemplars of varied performance levels
- Multimedia presentations
- Sheltered instruction
- Consultation with ESL teachers
- Manipulatives
- Tiered/Scaffolded Lessons
- Mnemonic devices
- Visual aids
- Modeling
- Guided note-taking
- Study Guides
- Modified homework
- Differentiated pre-typed class notes and example problems
- Use of the special education teacher to re-instruct in flexible small groups for the struggling learner
- Manipulatives
- Flipped Instruction
- Word banks
- Reduced choice on assessments
- Preferential seating
- Choice activities
- Modified time requirements
- Modified notes
- Modified lesson, assessment and study guide format
- Provide an enriched curriculum and activities
- Independent projects
- Contracts/behavior support plans
- Open-ended responses
- Project-based learning
- Group activities

- Guided Notes
- Functional learning incorporated into each lesson
- Exploration Activities
- Assessment read aloud
- Small group assessments
- Organizational Support
- Oral questioning assessments to supplement written response
- Pre-writing Structural Supports for extended writing tasks
- Ongoing teacher feedback as part of the writing process
- Interactive Study Guides
- Multi-sensory approach to instruction
- Written and spoken step-by-step directions
- Content-focused assessment (not grading for spelling/grammar)
- Graphic organizers
- Non-verbal cues to begin task/remain on task/refocus
- Individual monitoring for understanding/reinforced instruction
- Printed copies of class readings for application of Active Reading Strategies

Gifted & Talented

- Provide one-to-one teacher support
- Curriculum Compacting
- Advanced problems to extend the critical thinking skills of the advanced learner
- Supplemental reading material for independent study
- Elevated questioning techniques using Webb's Depth of Knowledge matrix
- Curriculum Compacting
- Flexible grouping
- Tiered assignments
- Topic selection by interest
- Manipulatives
- Tiered Lessons
- Flipped Instruction
- Multimedia Presentations
- Open-ended responses
- Project-based learning
- Group activities
- Guided Notes
- Conclusions and analysis of exploratory activities
- Career based learning incorporated into each lesson
- Exploration Activities

- Student choice

ELLs

- Exemplars of varied performance levels
- Multimedia presentations
- Sheltered instruction
- Consultation with ESL teachers
- Manipulatives
- Tiered/Scaffolded Lessons
- Mnemonic devices
- Visual aids
- Modeling
- Guided note-taking
- Study Guides
- Modified homework
- Differentiated pre-typed class notes and example problems
- Individualized instruction plans
- Manipulatives
- Flipped Instruction
- Words banks
- Reduced choice on assessments
- Preferential seating
- Choice activities
- Modified time requirements
- Modified notes
- Modify lesson, assessment and study guide format
- Provide an enriched curriculum and activities
- Contracts/management plans
- Open-ended responses
- Project-based learning
- Group activities
- Guided Notes
- Exploration Activities
- Assessment read aloud
- Small group assessments
- Oral questioning assessments to supplement written response
- Pre-writing Structural Supports for extended writing tasks
- Ongoing teacher feedback as part of the writing process
- Interactive Study Guides

- Multi-sensory approach to instruction
- Written and spoken step-by-step directions
- Graphic organizers
- Non-verbal cues to begin task/remain on task/refocus
- Individual monitoring for understanding/reinforced instruction
- Printed copies of class readings for application of Active Reading Strategies

504s

- Exemplars of varied performance levels
- Multimedia presentations
- Sheltered instruction
- Tiered/Scaffolded Lessons
- Mnemonic devices
- Visual aids
- Modeling
- Guided note-taking
- Study Guides
- Differentiated pre-typed class notes and example problems
- Manipulatives
- Words banks
- Reduced choice on assessments
- Preferential seating
- Modified time requirements
- Modified notes
- Modify lesson, assessment and study guide format
- Modified homework
- Independent projects
- Contracts/management plans
- Open-ended responses
- Project-based learning
- Group activities
- Guided Notes
- Exploration Activities
- Assessment read aloud
- Small group assessments
- Organizational Support
- Oral questioning assessments to supplement written response
- Pre-writing Structural Supports for extended writing tasks
- Ongoing teacher feedback as part of the writing process
- Interactive Study Guides
- Multi-sensory approach to instruction
- Written and spoken step-by-step directions
- Content-focused assessment (not grading for spelling/grammar)
- Graphic organizers

- Non-verbal cues to begin task/remain on task/refocus
- Individual monitoring for understanding/reinforced instruction
- Printed copies of class readings for application of Active Reading Strategies

Students at Risk of Failure

- Exemplars of varied performance levels
- Multimedia presentations
- Tiered/Scaffolded Lessons
- Modeling
- Guided note-taking
- Study Guides
- Differentiated pre-typed class notes and example problems
- Individualized instruction plans
- Words banks
- Reduced choice on assessments
- Preferential seating
- Choice activities
- Modified time requirements
- Modified notes
- Modified lesson, assessment and study guide format
- Modified homework
- Provide an enriched curriculum and activities
- Contracts/management plans
- Open-ended responses
- Project-based learning
- Group activities
- Guided Notes
- Exploration Activities
- Assessment read aloud
- Small group assessments
- Oral questioning assessments to supplement written response
- Pre-writing Structural Supports for extended writing tasks
- Ongoing teacher feedback as part of the writing process
- Interactive Study Guides
- Multi-sensory approach to instruction
- Written and spoken step-by-step directions
- Graphic organizers
- Non-verbal cues to begin task/remain on task/refocus
- Individual monitoring for understanding/reinforced instruction
- Printed copies of class readings for application of Active Reading Strategies

SCOPE AND SEQUENCE
COURSE OUTLINE/STUDENT OBJECTIVE

Unit 1: User Centered Design

Duration: 7 days

Overview: This unit introduces the design process as a specific version of the problem solving process in which empathy for a user's needs is consistently integrated. Students learn strategies for identifying user needs and assessing how well different designs address them. In particular they learn how to develop a paper prototype, how to gather and respond to feedback about a prototype, and consider ways different user interfaces do or do not affect the usability of their apps.

New Jersey Standards: 8.2.8.C.1; 8.2.8.C.2; 8.2.8.C.4; 8.2.8.C.5

21st Century: CRP2; CRP4; CRP6; CRP8; 9.3.ST-ET.3; 9.3.ST-SM.2; 9.3.IT-PRG.5; 9.3.IT-PRG.6; 9.3.IT-PRG.7

Cross-Curricular: RST.6-8.3; RST.6-8.4; RST.6-8.9; RST.6-8.10; MP.1;MP.2; MP.4; MS-ETS1-2; MS-ETS1-3; MS-ETS1-4

CSTA Standards

- 2-CS-01 - Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices..
- 2-CS-02 - Design projects that combine hardware and software components to collect and exchange data.
- 2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.
- 2-IC-21 - Discuss issues of bias and accessibility in the design of existing technologies.
- 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms.
- 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
- 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse.
- 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
- 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution.
- 2-AP-17 - Systematically test and refine programs using a range of test cases.
- 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
- 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.
- 2-IC-22 - Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.
- 2-DA-07 - Represent data using multiple encoding schemes.

- 2-DA-08 - Collect data using computational tools and transform the data to make it more useful and reliable.
- 2-DA-09 - Refine computational models based on the data they have generated.

Essential Questions:
- How do designers identify the needs of their user?
- How can we ensure that a user's needs are met by our designs?
- What processes will best allow us to efficiently create, test, and iterate upon our designs?

Student Learning Objectives:
*Students will know and be able t*o
- Express opinions respectfully and effectively
- Critically evaluate an object for how well its design meets a given set of needs
- Identify empathy for the user as an important component of the design process
- Distinguish between their own needs and the needs of their users
- Critique a design through the perspective of a user profile
- Design improvements to a product based on a user profile
- Empathize with a user's needs to design an object
- Create meaningful categories from a collection of ideas, specifically in the context of a brainstorm
- Use a paper prototype to test out an app before programming it
- Identify the user needs a prototype was designed to address
- Translate user needs into changes and improvements in the user interface of an app
- Categorize and prioritize user feedback for an app
- Create a paper prototype for the screens of an app
- Interview a peer to learn about their needs
- Analyze interview notes to develop follow-up questions
- Brainstorm potential solutions to a specific problem
- Design the functionality of an app to address the specific needs of a user
- Identify improvements to an app based on user testing
- Design the user interface of an app

Possible Activities
- The class explores a variety of different teapot designs to consider how design choices or made and why. Using the teapots as an example, the class will explore the relationship between users, their needs, and the design of objects they use.
- Using user profiles, the class explores how different users might react to a variety of products. Role playing as a different person, each member of the class will get to experience designs through someone else's eyes.
- In small groups, the class uses the design process to come up with ideas for smart clothing. From brainstorming, to identifying users, to finally proposing a design, this is the first of several opportunities in this unit to practicing designing a solution for the needs of others.
- To help out a developer with their idea, the class tests and provides an app prototype made of paper.

- Users have been testing an app, and they have lots of feedback for the developer. The class needs to sort through all of this feedback, identify the common themes and needs, and start revising the prototype to make it better meet the users' needs.
- The class will rely on each other as potential users, and pairs will get to interview each other to identify needs that could be addressed by developing an app.
- The class comes up with app ideas to address the needs of their users. To express those ideas, and test out their effectiveness, each student creates and tests paper prototypes of their own.

Unit 2: App Prototyping

Duration: 9 Class Meetings

Overview: This unit introduces the design process as a specific version of the problem solving process in which empathy for a user's needs is consistently integrated. Students learn strategies for identifying user needs and assessing how well different designs address them. In particular they learn how to develop a paper prototype, how to gather and respond to feedback about a prototype, and consider ways different user interfaces do or do not affect the usability of their apps.

New Jersey Standards: 8.2.8.D.1; 8.2.8.D.2; 8.2.8.D.3

21st Century: CRP2; CRP4; CRP6; CRP8; 9.3.ST-ET.3; 9.3.ST-SM.2; 9.3.IT-PRG.5; 9.3.IT-PRG.6; 9.3.IT-PRG.7

Cross-Curricular: RST.6-8.3; RST.6-8.4; RST.6-8.9; RST.6-8.10; MP.1;MP.2; MP.4; MS-ETS1-2; MS-ETS1-3; MS-ETS1-4

CSTA Standards

- 2-CS-01 - Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices..
- 2-CS-02 - Design projects that combine hardware and software components to collect and exchange data.
- 2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.
- 2-IC-21 - Discuss issues of bias and accessibility in the design of existing technologies.
- 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms.
- 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
- 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse.
- 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
- 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution.
- 2-AP-17 - Systematically test and refine programs using a range of test cases.
- 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
- 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.
- 2-IC-22 - Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.
- 2-DA-07 - Represent data using multiple encoding schemes.
- 2-DA-08 - Collect data using computational tools and transform the data to make it more useful and reliable.
- 2-DA-09 - Refine computational models based on the data they have generated.

Essential Questions:

- How do teams effectively work together to develop software?
- What roles beyond programming are necessary to design and develop software?
- How do designers incorporate feedback into multiple iterations of a product?

Student Learning Objectives:

*Students will know and be able t*o

- Identify ways in which apps can affect social change
- Locate apps that address a specific user group or need Identify the user needs being addressed by an app
- Communicate the design and intended use of program
- Demonstrate the user flow through an app's design using a paper prototype
- Test a prototype with a user, recording the results
- Analysing a user test to identify potential issues or improvements
- Translate a paper prototype into a digital format
- Select the appropriate input element for a given type of information
- Write programs that respond to user input
- Integrate screens designed by others into an app of their own
- Collaborate with others to develop an interactive prototype
- Write out a detailed plan for how they will test their low fidelity prototype with other people
- Run a user test on an app and record what users say about their minimum viable product
- Analyze the user feedback from the previous lesson and determine a list of bugs (flaws) that need to be fixed and features that could be added to the app
- Prioritize the bugs and features according to impact and ease of implementation
- Present technical information clearly to nontechnical users
- Reflect on the development of an ongoing project

Possible Activities

- The class organizes into teams and starts exploring app topics.
- Each group identifies a handful of apps that address the same topic they are working on, using those apps to help refine the app idea they will pursue.
- Student teams explore some example apps created in App Lab, using those apps to help inform the first paper prototypes of their apps.
- Students teams test out their paper prototypes with other members of the class. With one student role playing the computer, one narrating, and the rest observing, teams will get immediate feedback on their app designs which will inform the next version of their app prototypes.
- The teams now move to App Lab to build the next iteration of their apps. Using the drag-and-drop Design Mode, each team member builds out at least one page of their team's app, responding to feedback that was received in the previous round of testing.
- The teams combine screens into a single app. Simple code can then be added to make button clicks change to the appropriate screen.

- Teams run another round of user testing, this time with their interactive prototype. Feedback gathered from this round of testing will inform the final iteration of the app prototypes.
- Using the feedback from the last round of testing, teams implement changes that address the needs of their users. Each team tracks and prioritizes the features they want to add and the bugs they need to fix.
- Each team prepares a presentation to "pitch" the app they've developed. This is the time to share struggles, triumphs, and plans for the future.

Unit 3: Data and Society

Duration: 8 Class Meetings

Overview: In this unit, students focus on data representation and its role in solving information problems. Students learn what a representation system needs to be useful, and how computers are able to represent different types of information using binary systems. At the end of the unit, students represent their perfect day in a binary punch card and trade with classmates to decipher.

New Jersey Standards: 8.1.8.A.1; 8.2.8.E.3; 8.2.8.C.1

21st Century: CRP2; CRP4; CRP6; CRP8; 9.3.ST-ET.3; 9.3.ST-SM.2; 9.3.IT-PRG.5; 9.3.IT-PRG.6; 9.3.IT-PRG.7

Cross-Curricular: RST.6-8.3; RST.6-8.4; RST.6-8.9; RST.6-8.10; MP.1;MP.2; MP.4; MS-ETS1-2; MS-ETS1-3; MS-ETS1-4

CSTA Standards

- 2-DA-07 - Represent data using multiple encoding schemes.
- 2-DA-08 - Collect data using computational tools and transform the data to make it more useful and reliable.
- 2-NI-05 - Explain how physical and digital security measures protect electronic information.
- 2-NI-06 - Apply multiple methods of encryption to model the secure transmission of information.
- 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms.
- 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
- 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
- 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
- 2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.
- 2-IC-22 - Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.
  2-IC-23 - Describe tradeoffs between allowing information to be public and keeping information private and secure.

Essential Questions:

- Why is representation important in problem solving?
- What features does a representation system need to be useful?
- What is necessary to create usable binary representation systems?
- How can we combine systems together to get more complex information?

Student Learning Objectives:

- Define data as information collected from the world to help make a recommendation or solve a problem.
- Provide examples of how representing data in different ways can affect its ability to solve different problems.
- Choose the best way to represent some information based on how it will be used.
- Describe the necessary features of a system for representing information
- Create, use, and provide feedback on a system for representing information
- Iteratively improve upon a system for representing information by testing and responding to feedback
- Define a binary system as one that uses just two possible states to represent information
- Define a bit as a single piece of binary information
- Use the ASCII system to encode and decode text information in binary
- Create and manipulate binary patterns to represent black and white images
- Describe common features of systems used to represent information in binary
- Use a binary system to represent numbers.
- Extend a representation system based on patterns
- Apply a method of encryption to ensure the secure transmission of data.
- Use both physical and digital security measures to secure data.
- Use multiple binary systems to decode information.
- Determine the most appropriate encoding system for a given piece of information.
- Choose and justify the use of different binary representation systems depending on the information being represented
- Encode and decode information represented in binary numbers and ASCII text
- Create a generalized representation system for many instances of a complex type of information
-

Possible Activities

- Groups use a data set to make a series of meal recommendations for people with various criteria. Afterwards, groups compare their responses and discuss how the different representations of the meal data affected how the students were able to solve the different problems.
- Groups create systems that can represent any letter in the alphabet using only a single stack of cards, then create messages with their systems and exchange with other groups to ensure the system worked as intended. The class discusses commonalities between working systems while recognizing that there are many possible working solutions.
- The explores binary system and observe how computers represent information using either "on" or "off". The class is then introduced to the ASCII system for representing text using binary symbols and practices using this system before encoding their own messages using ASCII.
- The class is introduced to the concept of splitting images into squares or "pixels" which can then be turned on or off individually to make the entire image. After doing a short set of challenges using the Pixelation Widget, the class makes connections between the system for representing images and the system for representing text they learned in the previous lesson.

- With a set of cards that represent the place values in a binary (base-2) number system, the class turns bits "on" or "off" by turning cards face up and face down, then observes the numbers that result from these different patterns. Eventually, the pattern is extended to a generic 4-bit system.
- Students have a discussion on the different levels of security they would like for personal data. Once the class has developed an understanding of the importance of privacy, they learn about the process of encrypting information by enciphering a note for a partner and deciphering the partner's note. The class concludes with a discussion about the importance of both physical and digital security.
- After seeing a series of bits and being asked to decode them, the class is introduced to the idea that understanding binary information requires understanding both the system that is being used and the meaning of the information encoded.
- The class designs structure to represent their perfect day using the binary representation systems they've learned in this chapter. After deciding which pieces of information the record should capture, the class will decide how a punch card of bytes of information will be interpreted to represent those pieces of information. Afterwards, everyone will use the ASCII, binary number, and image formats they have learned to represent their perfect days try to decipher what a partner's perfect day is like.

Unit 4: Solving Data Problems

Duration: 9 Class Meetings

Overview: In this unit, students explore how data can be used to answer interesting questions and solve problems. Using a modified version of the general Problem Solving Process, students look at how computers and humans use data differently and the pros and cons of automating problem solving. After learning ways that computers use data in the real world, students choose their own problem and use data to address it.

New Jersey Standards: 8.2.8.C.7; 8.2.8.B.2; 8.2.8.C.4; 8.1.8.A.1; 8.1.8.A.3

21st Century: CRP2; CRP4; CRP6; CRP8; 9.3.ST-ET.3; 9.3.ST-SM.2; 9.3.IT-PRG.5; 9.3.IT-PRG.6; 9.3.IT-PRG.7

Cross-Curricular: RST.6-8.3; RST.6-8.4; RST.6-8.9; RST.6-8.10; MP.1;MP.2; MP.4; MS-ETS1-2; MS-ETS1-3; MS-ETS1-4

CSTA Standards

- 2-DA-07 - Represent data using multiple encoding schemes.
- 2-DA-08 - Collect data using computational tools and transform the data to make it more useful and reliable.
- 2-NI-05 - Explain how physical and digital security measures protect electronic information.
- 2-NI-06 - Apply multiple methods of encryption to model the secure transmission of information.
- 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms.
- 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
- 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
- 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
- 2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.
- 2-IC-22 - Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.
  2-IC-23 - Describe tradeoffs between allowing information to be public and keeping information private and secure.

Essential Questions:

- How does data help us to solve problems?
- How do computers and humans use data differently?
- What parts of the data problem solving process can be automated?
- What kinds of problems do computers use data to solve in the real world?

Student Learning Objectives:

- Use the problem solving process to answer a question using data.
- Identify and collect relevant data to help solve a problem.
- Use data to draw conclusions.
- Give examples of how data is collected from sensors and tracking user behavior.
- Determine data that would be helpful in solving a problem, and how that data could be collected.
- Distinguish between data that users intentionally and unintentionally produce.
- Identify and remove irrelevant data from a data set.
- Create a bar chart based on a set of data.
- Explain why a set of data must be cleaned before a computer can use it.
- Use tables and visualizations summarizing data to support a decision
- Present and critique interpretations of tables and visualizations.
- Identify additional data that could be collected to improve a decision
- Visually organize data to highlight relationships and support a claim.
- Use cross tabulation to find patterns and relationships in data.
- Design an algorithm for making decisions using data as inputs
- Explain the benefits and drawbacks of using computers for automated decision making
- Interpret collected data to identify patterns
- Apply the data problem solving process to a personally relevant topic
- Determine appropriate sources of data needed to solve a problem

Possible Activities

- The class is tasked with deciding what a city most needs to spend resources on. They must find and use data from the Internet to support their decision.
- The class looks at three scenarios that could be solved using data and brainstorms the types of data they would want to solve them and how they could collect the data. Each scenario also includes a video about a real-world service that has solved a similar problem with data.
- The class first looks at the how presenting data in different ways can help people to understand it better. After seeing how cleaning and visualization can help people make better decisions, the class looks at what parts of this process can be automated, and what needs a human.
- The students practice the data problem solving process and in some cases the class can and should decide that they should collect more data. The students engage in a discussion of how different people could draw different conclusions from the same data, or how collecting different data might have affected the decisions they made.
- Students begin the lesson by looking at a cake preference survey that allows respondents to specify both a cake and an icing flavor. They discuss how knowing the relationship between cake and icing preference helps them better decide which combination to recommend. They are then introduced to cross tabulation, which allows them to graph relationships to different preferences. They use this technique to find relationships in a preference survey, then brainstorm the different types of problems that this process could help solve.

- Students are given the task of creating an algorithm that could suggest a vacation spot. Students then create rules, or an algorithm, that a computer could use to make this decision automatically. Students share their rules and what choices their rules would make with the class data. They then use their rules on data from their classmates to test whether their rules would make the same decision that a person would. The lesson concludes with a discussion about the benefits and drawbacks of using computers to automate the data problem solving process.
- The class designs ways to use data to make a recommendations or predictions to help solve a problem. In the first several steps the class brainstorms problems, performs simple research, and defines a problem of their choosing. They then decide what kind of data they want to collect, how it could be collected, and how it could be used, before exchanging feedback and giving a final presentation.

SUGGESTED MATERIALS AND RESOURCES

Materials:

- Aluminum foil Plastic containers
- Poster paper
- Markers/colored pencils
- Scissors Glue and tape
- Packs of sticky notes
- Design Journals
- Desktop computer
- iPad apps and peripherals

Websites:

- https://studio.code.org/s/csd4-2018
- https://curriculum.code.org/csd-18/standards/
- www.code.org
- www.Youtube.com
- www.google.com/maps