Curriculum: Computer Science Principles (PLTW)

Curricular Unit: Introductory Programming with Scratch, MIT App Inventor and Python

Instructional Unit: A. Utilize the roles of variables to read and understand programs written in a variety of languages

**Standard Alignments (Section 2)**

<table>
<thead>
<tr>
<th>SCCLE: SC7.1.D</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAGLE: PP.1.C (HS Level 4)</td>
</tr>
<tr>
<td>Knowledge: (CA) 1,3,4,6 (MA) 1,5</td>
</tr>
<tr>
<td>CCSS: 9-10.RI.2; 9-10.W.10; 9-10.SL.1a; 9-10.SL.5; 9-10.SL.6; 9-10.L.1; 9-10.L.2c; 9-10.L.4a-d; 9-10.L.5b; 9-10.L.6; 9-10.RST.2; 9-10.RST.3; 9-10.RST.4; 9-10.WHST.10; A-SSE.1a;</td>
</tr>
<tr>
<td>NETS: 1b; 2a,c,d; 3a; 4b; 5b; 6</td>
</tr>
<tr>
<td>Performance: 1.4, 1.5, 1.10, 2.1, 2.5-2.7, 3.2-3.4, 3.7, 4.4-4.6, 4.8</td>
</tr>
</tbody>
</table>

**Unit (Section 3)**

**Learning Targets:**

- **Knowledge:**
  - Describe the role of creativity in designing an attractive, functional, and accessible graphical user interface
  - Describe a computer as responding to input in a deterministic manner that depends only on input and on the computer's state
  - Recognize common patterns employing variables, including value accumulation, list aggregation, and iteration across the elements of a collection
  - Describe ways to identify the existence and location of errors in software
  - Identify the decade in which milestones occurred in the development of computing and the Internet
  - Describe examples in which computation has or will create new societal phenomena and human capabilities to perceive and act upon our environment
  - Describe examples demonstrating that new ways to collaborate and share information are evolving
  - Describe career-oriented opportunities to use computational skills to positively affect people's lives
  - Describe how computing is connected to innovations in other fields

- **Skills:**
  - Create a visual artifact of a computational process, such as an image or screenshot of a user interface or web page
  - Communicate a design for a program using natural language, diagrams, and pseudocode
  - Design a program by breaking a large plan into smaller modules
  - Utilize the tools incorporated in an IDE and/or higher-level language to create original programming solutions
  - Create a program by incrementally writing and testing modular code
• Improve readability, efficiency, or correctness of code from other people
• Create readable code with a combination of documentation and comments
• Deduce the value of a variable resulting from execution of particular code
• Document a software development process
• Document a software product using high-level documentation
• Effectively manage a project, including planning and time management, team norming and load balancing, file/revision management, and documentation
• Collaborate effectively with others when managing a project
• Compare to select from among several tools available for collaboration
• Collaborate when programming
• Collaborate when presenting
• Communicate an idea for a product that solves a problem or expresses creativity
• Communicate the progress on a project, including accomplishments and next steps

Instructional Strategies:

• Project Based Learning will be implemented in the following ways – The teacher will:
  • model a variety of skills using different programs including:
    • Scratch
    • MIT App Inventor
    • Canopy
    Students will practice those skills and utilize them to create more complex codes
  • facilitate collaboration between pairs to create complex codes and critical thinking exercises
  • create instructional videos to demonstrate coding skills. Students will utilize:
    • direct instruction
    • pair programming
    • instructional videos
    to complete a variety of coding tasks
  • Given teacher-created scenarios, students will create and share programs that can be utilized in real world scenarios

Assessments/Evaluations:

• Formative (assessed using scoring guides, observation, and practice activities) – Students will produce a program to play a game or tell a story in order to:
  • express themselves creatively
  • use algorithms and operations appropriately
  • decompose problems and maintain persistence
  • document and present their creation
  • write about the impact of computing to describe how computing is changing:
    • society
    • most fields of work
- Summative (assessed using scoring guides, observation, and practice activities) – Students will:
  - collaborate when developing software to demonstrate helpful practices when collaborating
  - explain and improve code using:
    - operations
    - variables
    - algorithms

### Sample Assessment Questions:

- How do computers perform complicated tasks built from simple instructions?
- How are variables used in programming?
- How do programmers approach a complicated problem?
- What role does creativity play in algorithmic programming?
- What makes for a good process for collaborative software development?

### Instructional Resources/Tools:

- A1.1.1 – Principles
- A1.1.2 – Lightbot - Input, State, Output
- A1.1.3 – Scratch Graphics and If-Else Blocks
- A1.1.4 – Scratch Objects and Methods
- A1.1.5 – Scratch Variable Roles Part I
- A1.1.6 – Scratch Iteration and Variable Roles Part II
- B1.1.7 – Scratch Game or Story

### Cross Curricular Connections:

- ELA:
  - Reading
  - Writing
  - Speaking and listening
- Math:
  - Arithmetic
  - Algebraic expressions
  - Logic problems

### Depth of Knowledge (Section 5)

DOK: 4
Curriculum: Computer Science Principles (PLTW)

Curricular Unit: Introductory Programming with Scratch, MIT App Inventor and Python

Instructional Unit: B. Develop a computational algorithmic solution in a variety of contexts and disciplines

**Standard Alignments (Section 2)**

<table>
<thead>
<tr>
<th>SCCLE: SC7.1.D</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAGLE: PP.1.C (HS Level 4)</td>
</tr>
<tr>
<td>Knowledge: (CA) 1,3,4,6 (MA) 1,5</td>
</tr>
<tr>
<td>NETS: 1b; 2a,c,d; 3a; 4b; 5b; 6</td>
</tr>
<tr>
<td>Performance: 1.4, 1.5, 1.10, 2.1, 2.5-2.7, 3.2-3.4, 3.7, 4.4-4.6, 4.8</td>
</tr>
</tbody>
</table>

**Unit (Section 3)**

**Learning Targets:**

- **Knowledge:**
  - Describe the role of creativity in designing a mobile application to solve a problem
  - Describe the ways in which various types of digital data can be represented in binary
  - Distinguish continuous and discrete phenomena and identify digital and analog data
  - Describe layers of abstraction that help people represent and use data
  - Identify layers of abstraction used in programming languages and describe advantages and disadvantages inherent in working at a high level of abstraction
  - Describe the implications of the limited precision of digital information in applications
  - Recognize events and event-handlers implied by a user interface

- **Skills:**
  - Create a graphical user interface
  - Consider implications of converting data from one representation to another, for example noise or compression
  - Convert a given quantity among bases 2, 10, and 16
  - Analyze and test code from other people
  - Extend, or apply to new purpose, code from other people
  - Encapsulate a set of related statements in a function or procedure
### Instructional Strategies:

- **Project Based Learning** will be implemented in the following ways – The teacher will:
  - model a variety of skills using different programs including:
    - Scratch
    - MIT App Inventor
    - Canopy
  - Students will practice those skills and utilize them to create more complex codes
  - facilitate collaboration between pairs to create complex codes and critical thinking exercises
  - create instructional videos to demonstrate coding skills. Students will utilize:
    - direct instruction
    - pair programming
    - instructional videos
  - to complete a variety of coding tasks
- **Given** teacher-created scenarios, students will create and share programs that can be utilized in real world scenarios

### Assessments/Evaluations:

- **Formative** – (assessed using scoring guides, observation, and practice activities) Students will identify a problem or need and develop an app to address the problem or need and will:
  - show creativity when solving a problem
  - document a design process and product
  - collaborate and reflect on collaboration
  - design or select and implement algorithms to create an app
  - explain the algorithms they have developed
  - present a problem and the progress and next steps during development of a solution
  - demonstrate problem decomposition, persistence, and iterative development
- **Summative** (assessed using scoring guides, observation, and practice activities) – Students will:
  - represent numeric, text, and color data with binary sequences, and explain the abstraction represented by using a digital representation to correctly represent it
  - contrast development environments and programming languages to describe strengths and weaknesses of tools
  - explain the abstraction of a function or procedure and its arguments and discuss modularity and reusability as strategies for handling complexity
  - write about future opportunities created by computing and consider:
    - the impact of networked, mobile and embedded computing
    - career opportunities
  - analyze code and identify patterns in which an algorithm uses variables

### Sample Assessment Questions:

- What do programming languages and development environments have in common?
- What can be represented by binary data?
- What contributes to an effective process for software development?
- How can a program be analyzed, understood, and modified?
Instructional Resources/Tools:

- A1.2.1 – Bits and Bytes
- A1.2.2 – Introducing App Inventor
- A1.2.3 – Creating An App
- A1.2.4 – Analyzing A Program
- P1.2.5 – Modifying A Program
- B1.2.6 – Designing an App

Cross Curricular Connections:

- ELA:
  - Reading
  - Writing
  - Speaking and listening
- Math:
  - Arithmetic
  - Algebraic expressions
  - Logic problems

Depth of Knowledge (Section 5)

DOK: 4
Curriculum: Computer Science Principles (PLTW)

Curricular Unit: Introductory Programming with Scratch, MIT App Inventor and Python

Instructional Unit: C. Develop proficiency learning in unfamiliar programming language

**Standard Alignments (Section 2)**

| SCCLE: SC7.1.D; SC8.1.B  
| VAGLE: PP.1.C (HS Level 4)  
| Knowledge: (CA) 1,3,4,6 (MA) 1,5  
| CCSS: 9-10.W.10; 9-10.SL.1; 9-10.SL.6; 9-10.L.1; 9-10.L.2c; 9-10.L.4a-d; 9-10.L.5b; 9-10.L.6; 9-10.RST.3; 9-10.RST.4; 9-10.RST.7; 9-10.WHST.10; S-MD.5; S-MD.7  
| NETS: 1b; 2a,c,d; 3a; 4b; 5b; 6  
| Performance: 1.4, 1.5, 1.10, 2.1, 2.5-2.7, 3.2-3.4, 3.7, 4.4-4.6, 4.8 |

**Unit (Section 3)**

**Learning Targets:**

- **Knowledge:**
  - Describe the ways in which various types of digital data can be represented in binary
  - Describe layers of abstraction that help people represent and use data
  - Identify layers of abstraction used in programming languages and describe advantages and disadvantages inherent in working at a high level of abstraction
  - Describe a computer as responding to input in a deterministic manner that depends only on input and on the computer's state
  - Recognize common patterns employing variables, including value accumulation, list aggregation, and iteration across the elements of a collection
  - Describe ways to identify the existence and location of errors in software
  - Describe the role that functions play in developing software
  - Distinguish among a variety of educational and reference resources related to code libraries
  - Describe a version control system
  - Describe the impact that computing has had in the social sciences, geography, and civics

- **Skills:**
  - Create a visual artifact of a computational process, such as an image or screenshot of a user interface or web page
  - Analyze a computational artifact for usability with a specific audience in mind
  - Consider implications of converting data from one representation to another, for example noise or compression
  - Communicate a design for a program using natural language, diagrams, and pseudocode
  - Evaluate a program for efficiency
- Design a program by breaking a large plan into smaller modules
- Analyze and test code from other people
- Extend, or apply to new purpose, code from other people
- Explore and use documentation and public information to extend the student's own knowledge of a programming language or to achieve a computational approach to solve a problem
- Utilize the tools incorporated in an IDE and/or higher-level language to create original programming solutions
- Evaluate programs written by others for readability
- Create a program by incrementally writing and testing modular code
- Identify appropriate boundary conditions for testing a program
- Improve readability, efficiency, or correctness of code from other people
- Create readable code with a combination of documentation and comments
- Encapsulate a set of related statements in a function or procedure
- Create a function to perform a particular calculation from the function's arguments
- Deduce the value of a variable resulting from execution of particular code
- Work with a group to agree on a policy or protocol
- Document a software development process
- Utilize a version control system
- Effectively manage a project, including planning and time management, team norming and load balancing, file/revision management, and documentation
- Collaborate effectively with others when managing a project
- Compare to select from among several tools available for collaboration
- Collaborate when programming
- Collaborate when presenting
- Present original research and research from literature
- Distinguish sources of information and comment on the information's reliability and intended audience
- Communicate an idea for a product that solves a problem or expresses creativity
- Communicate the progress on a project, including accomplishments and next steps

**Instructional Strategies:**

- Project Based Learning will be implemented in the following ways – The teacher will:
  - model a variety of skills using different programs including:
    - Scratch
    - MIT App Inventor
    - Canopy
  - facilitate collaboration between pairs to create complex codes and critical thinking exercises
  - create instructional videos to demonstrate coding skills. Students will utilize:
    - direct instruction
    - pair programming
    - instructional videos
  to complete a variety of coding tasks
- Given teacher-created scenarios, students will create and share programs that can be utilized in real world scenarios

**Assessments/Evaluations:**

- **Formative (assessed using scoring guides, observation, and practice activities)** – Students will create a function to implement an algorithm that carries out an intelligent strategy in a multi-player game in which the opponent's psyche must be considered. The algorithm is informed by the game history between the strategy algorithm itself and the opponent. In doing so, students will:
  - implement an algorithm for complex strategy
  - organize code well, with functions to support decomposition, reuse, and abstraction
  - demonstrate persistence and an iterative development process
  - explain a strategy and the algorithm created to implement a strategy
  - effectively collaborate
  - use an agreed upon process to version control collaborative work

- **Summative (assessed using scoring guides, observation, and practice activities)** – Students will:
  - describe a processor's execution of instructions by describing:
    - the fetch-execute cycle of one processor core
    - how programs run on a multicore processor
  - create code and:
    - provide corresponding descriptions of an algorithm in natural language, pseudocode, and code
    - create modular solutions, demonstrated by creation of functions and problem decomposition
    - use various data types
  - analyze and improve another’s code and:
    - recognize patterns employing variables
    - improve readability
  - develop code collaboratively and will:
    - show effectiveness of collaborative approach
    - debug and decompose code with persistence
    - agree on and use version control system
    - reflect on collaboration
  - correctly compare two algorithms in terms of equivalency and speed
  - describe career opportunities created by or impacted by computing and will:
    - show awareness and interest in the impact of computing on careers
    - describe computing work as collaborative

**Sample Assessment Questions:**

- What can be represented by binary data?
- How does abstraction make the software development process easier?
- What are the practices that lead to effective collaboration?
- What role does creativity play in algorithmic programming?
Instructional Resources/Tools:

- A1.3.1 – Instructions as Data
- A1.3.2 – Python Variables and Functions
- A1.3.3 – Branching and Output
- A1.3.4 – Nested Branching and Input
- A1.3.5 – Collections with Strings
- A1.3.6 – Collections with Tuples and Slicing
- A1.3.7 – Iteration with For Loops
- A1.3.8 – Iteration with While Loops
- A1.3.9 – Tools for Collaboration
- P1.3.10 – Prisoner's Dilemma

Cross Curricular Connections:

- ELA:
  - Reading
  - Writing
  - Speaking and listening
- Math:
  - Arithmetic
  - Algebraic expressions
  - Logic problems

**Depth of Knowledge (Section 5)**

DOK: 4
Curriculum: Computer Science Principles (PLTW)

Curricular Unit: Introductory Programming with Scratch, MIT App Inventor and Python

Instructional Unit: D. Select and use libraries of code written by other people by referring to documentation

**Standard Alignments (Section 2)**

| SCCLE: | SC7.1.D; SC8.1.B |
| VAGLE: | PP.1.C (HS Level 4) |
| Knowledge: | (CA) 1,3,4,6 (MA) 1,5 |
| NETS: | 1b; 2a,c,d; 3a; 4b; 5b; 6 |
| Performance: | 1.4, 1.5, 1.10, 2.1, 2.5-2.7, 3.2-3.4, 3.7, 4.4-4.6, 4.8 |

**Unit (Section 3)**

**Learning Targets:**

- **Knowledge:**
  - Articulate a range of positions on the question of ownership of bits
  - Describe what metadata contain and how they can be used
  - Discuss societal implications of the persistence and the ease of copying digital information

- **Skills:**
  - Work with a file system tree

**Instructional Strategies:**

- Project Based Learning will be implemented in the following ways – The teacher will:
  - model a variety of skills using different programs including:
    - Scratch
    - MIT App Inventor
    - Canopy
  - Students will practice those skills and utilize them to create more complex codes
  - facilitate collaboration between pairs to create complex codes and critical thinking exercises
• create instructional videos to demonstrate coding skills. Students will utilize:
  • direct instruction
  • pair programming
  • instructional videos
to complete a variety of coding tasks
• Given teacher-created scenarios, students will create and share programs that can be utilized in real world scenarios

Assessments/Evaluations:

• Formative – Students will describe an algorithm in natural language, psuedocode, and code that could be used to modify an image. Script the implementation of that algorithm across a set of images and:
  • create a software solution to a problem
  • decompose a problem and show persistence while solving the component sub-problems
  • document the development of a solution and create readable code
  • collaborate with team members and reflect upon collaboration
  • present a product and development process
  • present abstractions that were created or used
• Summative (assessed using scoring guides, observation, and practice activities) – Students will:
  • discuss and write about abstraction:
    • Use UML to describe methods and attributes of a class
    • Use prose to describe concepts of class, object, attributes, and methods
  • manipulate images using iteration:
    • Represent images as RGBpixel arrays
    • Use lists and arrays as data abstractions
  • work with file system trees: Navigate a file system and examine filenames and metadata
  • manipulate images using object-oriented method calls:
    • use an object-oriented API
    • learn from documentation
    • iterate an object.method() call across a list of objects
    • demonstrate effective teamwork
    • demonstrate helpful programming practices
  • read, write, and speak about societal issues involving digital data:
    • Analyze short- and long-term beneficial and harmful impacts of a computing innovation; create a professional technical report and a creative visual artifact about the innovation
    • Argue for greater or lesser property rights
    • Place the present in a historical context of rapid change
    • Describe issues around persistence of data and privacy

Sample Assessment Questions:

• What can be represented by binary data?
• How does abstraction make the software development process easier?
• What are the practices that lead to effective collaboration?
• What role does creativity play in algorithmic programming?
**Instructional Resources/Tools:**

- A1.4.1 – Procedural Abstraction
- A1.4.2 – Images as Files and Objects
- A1.4.3 – Images as Arrays
- A1.4.4 – Python Imaging Library API
- P1.4.5 – Algorithms for Images
- A1.4.6 – Digital Property and Forensics
- B1.4.7 – Image Artist

**Cross Curricular Connections:**

- **ELA:**
  - Reading
  - Writing
  - Speaking and listening
- **Math:**
  - Arithmetic
  - Algebraic expressions
  - Logic problems

---

**Depth of Knowledge (Section 5)**

DOK: 4
Curriculum: Computer Science Principles (PLTW)

Curricular Unit: Introductory Programming with Scratch, MIT App Inventor and Python

Instructional Unit: E.
- Select tools and use documentation to develop skill with a tool
- Create an effective user interface for a computational problem related to the students’ own interests and needs
- Design solutions with human-centric considerations
- Solve problems by breaking them down into modular parts
- Approach problems with persistence, curiosity, and an iterative approach

Standard Alignments (Section 2)

<table>
<thead>
<tr>
<th>SCCLE: SC7.1.D; SC8.1.B</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAGLE: PP.1.C (HS Level 4)</td>
</tr>
<tr>
<td>Knowledge: (CA) 1,3,4,6 (MA) 1,5</td>
</tr>
<tr>
<td>CCSS: 9-10.W.10; 9-10.SL.1c; 9-10.SL.5; 9-10.SL.6; 9-10.L.1; 9-10.L.2c; 9-10.L.4a-d; 9-10.L.5b; 9-10.L.6; 9-10.RST.3; 9-10.RST.4; 9-10.WHST.10; G-MG.1; G-MG.3</td>
</tr>
<tr>
<td>NETS: 1b; 2a,c,d; 3a; 4b; 5b; 6</td>
</tr>
<tr>
<td>Performance: 1.4, 1.5, 1.10, 2.1, 2.5-2.7, 3.2-3.4, 3.7, 4.4-4.6, 4.8</td>
</tr>
</tbody>
</table>

Unit (Section 3)

Learning Targets:
- Knowledge:
  - Describe the model-view-controller pattern and relate to a particular software solution
  - Recognize that a solution to one problem, such as a particular sorting or optimization task, can be used to solve seemingly dissimilar problems
  - Identify fields of computing careers

Instructional Strategies:
- Project Based Learning will be implemented in the following ways – The teacher will:
  - model a variety of skills using different programs including:
    - Scratch
    - MIT App Inventor
    - Canopy
    - Students will practice those skills and utilize them to create more complex codes
  - facilitate collaboration between pairs to create complex codes and critical thinking exercises
  - create instructional videos to demonstrate coding skills. Students will utilize:
    - direct instruction
    - pair programming
    - instructional videos
to complete a variety of coding tasks
• Given teacher-created scenarios, students will create and share programs that can be utilized in real-world scenarios

Assessments/Evaluations:

• Formative (assessed using scoring guides, observation, and practice activities) – Students will:
  • create a user interface with code:
    • Use operations and data/procedural abstractions
    • Create modular, commented code reflecting decomposition
    • Document a project with iterative cycles of problem definition and creative development with version control
  • Collaborate
  • Reflect
  • Present
• evaluate a user interface and suggest improvements for usability and accessibility:
  Evaluate and suggest improvements that demonstrate knowledge of usability criteria
• Summative (assessed using scoring guides, observation, and practice activities) – Students will:
  • communicate orally and in writing ideas about human-computer interfaces:
    • Give examples in which computing extends perception and ability to act
    • Give examples of ways a user interface excludes people or might be changed to include people
  • Write about an HCI job
  • Write about events and event handlers that are inferred from a user interface
• modify and create code:
  • Fix bugs and use API functions from documentation
  • Appropriately apply operations to object attributes
  • Collaborate
  • Iterate a function call, passing the function a variable related to the iteration
• explain an interface and the data behind it using the model-view-controller pattern: Identify the model, view, and controller design elements in an interface

Sample Assessment Questions:

• How does abstraction make the software development process easier?
• What are the practices that lead to effective collaboration?
• What role does creativity play in algorithmic programming?
• How is computing affecting the way we live our lives?
• How will computing change our world?

Instructional Resources/Tools:

• A1.5.1 – Human-Computer Interaction
• A1.5.2 – The API for the Tkinter Canvas
• A1.5.3 – The MVC Pattern with Tkinter
• B1.5.4 – Design a Python GUI
Cross Curricular Connections:

- ELA:
  - Reading
  - Writing
  - Speaking and listening
- Math:
  - Arithmetic
  - Algebraic expressions
  - Logic problems

**Depth of Knowledge (Section 5)**

DOK: 4
Curriculum: Computer Science Principles (PLTW)
Curricular Unit: Exploring Networking, the Internet, and the Web
Instructional Unit: F. Conduct oneself safely on the Internet in business and personal contexts, with respect to privacy and protection of assets

**Standard Alignments (Section 2)**

| VAGLE: PP.1.C (HS Level 4) |
| Knowledge: (CA) 1,3,4,6 |
| CCSS: 9-10.RI.1; 9-10.RI.2; 9-10.RI.4; 9-10.RI.6; 9-10.RI.8; 9-10.RI.10; 9-10.SL.1a,c; 9-10.SL.5; 9-10.SL.6; 9-10.L.1; 9-10.L.2c; 9-10.L.3; 9-10.L.4a-d; 9-10.5b; 9-10.L.6; 9-10.RH.4; 9-10.RH.5; 9-10.RH.10; 9-10.RST.3; 9-10.RST.4; 9-10.WHST.6; 9-10.WHST.10 |
| NETS: 1a; 2a,c,d; 3a-c; 4c; 5 |
| Performance: 1.1, 1.2, 1.4, 1.7, 1.10, 2.1-2.3, 2.5, 2.7, 3.1-3.3, 3.7, 4.5, 4.6, 4.8 |

**Unit (Section 3)**

**Learning Targets:**

- **Knowledge:**
  - Identify the relationship among nodes in a tree, as applied to the DOM in a Web page
  - Identify syntactic elements of HTML and CSS
  - Describe the hierarchical nature of the domain name system and IPv4 and IPv6 numbering systems
  - Describe the dependence of DNS on certificate authorities
  - Describe abstractions that enable the Internet to function, including IP addressing and domain name service
  - Describe the purpose and general nature of protocols that enable the Internet to function, including DNS, TCP/IP, SMTP, and HTTP protocols
  - Describe the governance of the Internet and the organizations that develop and maintain relevant standards
  - Identify examples of redundancy and autonomy in the physical and software systems of the Internet
  - Explain how redundancy and autonomy make the systems of the Internet scalable
  - Explain an encryption system
  - Describe the combination of encryption protocols with other protocols to provide secure transfer of information
  - Characterize the size, cost, and speed of computational processing as changing exponentially
  - Distinguish content from style and explain how accessibility requires a separation of these concerns
  - Analyze access to the Internet and to data among people in different countries, and connect to measures of health and wealth
• Contrast the patterns of inequity characterizing the personal computing revolution and the current mobile computing revolution
• Describe how specific career fields, e.g., biology and marketing, have changed profoundly as a result of automated collection and processing of data

Skills:
• Design, create, and publish a basic web page referencing a style sheet
• Calculate time, file size, and bandwidth given two of the three
• Navigate a file system tree
• Connect the rendering of a Web page with corresponding elements of HTML and CSS source code and with client browser settings
• Identify HTML, CSS, JavaScript, PHP, or SQL as an appropriate language for a particular task
• Use appropriate tools to manage files on a server permitting FTP
• Use appropriate tools to observe bandwidth and latency
• Analyze a URL to identify protocol, the host and domain names, the directory path, the filename, and the query string content, and describe the information implied about the nature of the requested content and its publisher
• Contrast search results provided by competing search algorithms to identical queries
• Use appropriate tools to observe IP addressing, IP routing, and DNS resolution

Instructional Strategies:
• The teacher will:
  • model safety precautions while using the Internet.
  • facilitate:
    • the collaboration of pair programming
    • discussions about the:
      • implications the Internet has had on society
      • importance of using safe practices when using the Internet
    • demonstrate the process of how the Internet handles information and how users access it

Assessments/Evaluations:
• Formative (assessed using scoring guides, observation, and practice activities) – Students will create a small set of linked web pages and:
  • use HTML tags
  • apply CSS styles
  • navigate a remote file system through a command line interface
  • use reference materials to learn a language
• Summative (assessed using scoring guides, observation, and practice activities) – Students will:
  • describe the impact and pace of change caused by networked computing and describe:
    • distributed computing
    • beneficial and harmful impacts of a computing innovation
    • the rate at which the impacts of computing have been and are developing
• analyze a network connection: Use tools to investigate:
  • DNS
  • TCP/IP
  • bandwidth
  • latency
• analyze the reliability of a web page – Infer information:
  • from a URL
  • about author and audience from a web page’s content
• relate search engine results to both the query and the search engine:
  • Skillfully construct a search engine query
  • Describe results as affected by censorship, optimization tactics, and engine algorithms
• describe hardware components of a computer and describe the:
  • processor
  • storage
  • RAM
  • NIC
  among the hardware components of a computer
• explain protocols by which entering a URL in a browser results in a rendered page:
  • Describe the role of TCP/IP in retrieving a web page
  • Describe the role of DNS in retrieving a web page
  • Generalize the concept of a "protocol" and relate to abstraction
  • Explain encryption, including the use of paired key authentication
  • Create examples in the languages of HTML and CSS
  • Relate HTML and CSS to accessibility

Sample Assessment Questions:

• How does the Internet work?
• How can we protect ourselves, our privacy, and our assets when working on the Internet?
• How has the Internet affected society?

Instructional Resources/Tools:

• A2.1.1 – The Rise of the Internet
• A2.1.2 – Your Favorite Web Page
• A2.1.3 – Protocols and Bandwidth
• A2.1.4 – HTML and CSS
• A2.1.5 – Secure Protocols

Cross Curricular Connections:

• ELA:
  • Reading
  • Writing
  • Speaking and listening
• Math:
  • Arithmetic
  • Algebraic expressions
  • Logic problems
### Depth of Knowledge (Section 5)

| DOK: 4 |
Curriculum: Computer Science Principles (PLTW)

Curricular Unit: Exploring Networking, the Internet, and the Web

Instructional Unit: G. Create interactive web sites to meet individual needs and interests

Standard Alignments (Section 2)

| SCCLE: | SC8.1.B |
| VAGLE: | PP.1.C (HS Level 4) |
| Knowledge: | (CA) 1,3,4,6 |
| NETS: | 1a,b; 2a,c,d; 3a; 4a,b,d; 5b; 6 |

Performance: 1.1, 1.2, 1.4, 1.7, 1.10, 2.1-2.3, 2.5, 2.7, 3.1-3.3, 3.7, 4.5, 4.6, 4.8

Unit (Section 3)

Learning Targets:

- Knowledge:
  - Describe the structure of a relational database
  - Describe how mathematical functions and algorithms can be applied to a data set to construct a derived data set
  - Describe the reasons for the rise in non-relational databases
  - Identify syntactic elements of JavaScript, PHP, and SQL

- Skills:
  - Create and publish a basic web page containing JavaScript
  - Describe the digital footprint left behind and analyze the implications for privacy that result from various actions conducted in a Web browser, including DNS requests, content requests, third-party content requests, cookies, and cached content (U6.06)
  - Identify vulnerabilities to social engineering, including phishing and the delivery of viruses through various file types, and respond appropriately
  - Identify an appropriate language for serving a particular purpose in web serving, including tasks appropriate to client-side and server-side scripting

Instructional Strategies:

- Project-Based Learning will be implemented
- The teacher will:
  - facilitate the collaboration of pair programming
  - model a variety of methods for creating and displaying information on the Web
  - guide students to explore a variety of social media outlets and their uses
Assessments/Evaluations:

- Formative (assessed using scoring guides, observation, and practice activities) – Students will create a data-driven website and:
  - design an appealing and usable interface
  - explore HTML5 and JavaScript functionality to develop the interface
  - use a relational database and create SQL queries to develop the interface
  - explore a variation in PHP code to develop the interface
  - collaborate to create a program using a blend of web client-side and web server-side code
  - present the design for a solution to a client's need

- Summative (assessed using scoring guides, observation, and practice activities) – Students will:
  - describe CS, IT, and Web developer roles and the human creativity that produces the Web and describe:
    - possible career paths in:
      - CS
      - IT
      - database management
      - web development
      - the Internet's governance
  - create a web page using JavaScript
  - analyze a web page and:
    - Describe how:
      - HTML
      - CSS
      - JavaScript
      produce a rendered page
    - Distinguish the roles and mechanics of server-side and client-side scripting
  - describe a digital footprint: Students will describe a footprint left behind by a user during Web browsing
  - analyze code and:
    - recognize patterns using variables
    - compare the time efficiencies of two programming approaches to a problem

Sample Assessment Questions:

- How does the Internet work?
- How can we protect ourselves, our privacy, and our assets when working on the Internet?
- How has the Internet affected society?

Instructional Resources/Tools:

- A2.2.1 – HTML5 and JavaScript
- A2.2.2 – Introducing PHP
- A2.2.3 – Databases and SQL
- P2.2.4 – Dynamic Data-Driven Design
- A2.2.5 – Career Fields of CS and IT
Cross Curricular Connections:

- ELA:
  - Reading
  - Writing
  - Speaking and listening
- Math:
  - Arithmetic
  - Algebraic expressions
  - Logic problems

Depth of Knowledge (Section 5)

DOK: 4
Curriculum: Computer Science Principles (PLTW)

Curricular Unit: Exploring Networking, the Internet, and the Web

Instructional Unit: H. Analyze methods of encryption and forms of cyber security

**Standard Alignments (Section 2)**

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge: (CA)</td>
<td>1,3,4,6</td>
</tr>
<tr>
<td>CCSS:</td>
<td>9-10.RI.1; 9-10.RI.2; 9-10.RI.4; 9-10.RI.6; 9-10.RI.8; 9-10.W.1a-e; 9-10.W.2a-f; 9-10.W.4; 9-10.W.5; 9-10.W.6; 9-10.W.7; 9-10.W.8; 9-10.W.9; 9-10.W.10; 9-10.SL.1a-d; 9-10.SL.2; 9-10.SL.3; 9-10.SL.4; 9-10.SL.6; 9-10.L.1a,b; 9-10.L.2c; 9-10.L.3a; 9-10.La-d; 9-10.L.5b; 9-10.L.6; 9-10.RH.1; 9-10.RH.2; 9-10.RH.4; 9-10.RH.5; 9-10.RH.6; 9-10.RH.9; 9-10.RH.10; 9-10.RST.1; 9-10.RST.2; 9-10.RST.3; 9-10.RST.4; 9-10.RST.6; 9-10.RST.7; 9-10.RST.9; 9-10.RST.10; 9-10.WHST.1a,c-e; 9-10.WHST.2a-f; 9-10.WHST.4; 9-10.WHST.5; 9-10.WHST.7; 9-10.WHST.8; 9-10.WHST.9; 9-10.WHST.10; F-IF.6; F-L.E.1b,c; F-L.E.3; S-ID.6a; S-ID.7</td>
</tr>
<tr>
<td>NETS:</td>
<td>1a,b; 2a,c,d; 3a; 4a,b,d; 5b; 6</td>
</tr>
<tr>
<td>Performance:</td>
<td>1.1, 1.2, 1.4, 1.7, 1.10, 2.1-2.3, 2.5, 2.7, 3.1-3.3, 3.7, 4.5, 4.6, 4.8</td>
</tr>
</tbody>
</table>

**Unit (Section 3)**

**Learning Targets:**

- **Knowledge:**
  - Identify the relationship between parallel computation and computationally intensive tasks like simulation, modeling, and analysis of large data sets
  - Estimate the time for an algorithm to operate on a data set, given the algorithm's running time on another data set and the algorithm's complexity in big-O notation
  - Explain the role of prime numbers and tractability in encryption
  - Describe the mechanism of any particular cyber attack
  - Identify measures that contribute to an effective defense from any particular cyber attack
  - Identify the roles of software developers, government, industry, employees, standards bodies, consumers, and citizens in securing information and processing power
  - Describe codes of ethics and professional conduct for cyber security professionals

- **Skills:**
  - Analyze the security of situations in which a user is prompted to follow a link, download content, or provide information in email and Web forms
  - Use appropriate tools and techniques to implement defensive cyber measures
Instructional Strategies:

- The teacher will:
  - facilitate:
    - the collaboration of pair programming
    - discussion about the importance of cyber security
  - present a variety of digital security and encryption methods
  - guide students to determine the appropriate methods of digital security for a variety of real-world scenarios

Assessments/Evaluations:

- Formative (assessed using scoring guides, observation, and practice activities) – Students will:
  - protect oneself online and:
    - identify cyber defense measures to take when using a browser and when using email
    - relate design properties of operating systems, the Internet, and Web and email services to vulnerabilities and defenses
    - describe the boundary between ethical behavior and unethical hacking and describe consequences for unethical hacking
    - describe career opportunities helping securing people's cyber systems
    - reverse-engineer code, decrypt data, and identify functionality and disfunctionality of code not overtly stated by another programmer
  - Succeed in a national capture-the-flag competition

- Summative (assessed using scoring guides, observation, and practice activities):
  - Compare two algorithms:
    - Compare algorithms theoretically and make empirical measurements
    - Relate algorithms and empirical data to orders of time complexity
    - Enrichment: Relate the comparison to:
      - P/NP
      - computability
      - NP complete problems
  - Enrichment: Explain RSA encryption: Explain the mathematical procedure and relate the size of the prime numbers to the time required for brute force decryption
  - Describe how networked computing impacts government:
    - Describe consequences of networked computing for privacy, law enforcement, and democracy
    - Analyze short- and long-term beneficial and harmful impacts of a computing innovation; create a professional technical report and a creative visual artifact about the innovation

Sample Assessment Questions:

- What is the nature of attack and defense in cyber security?

Instructional Resources/Tools:

- A2.3.1 – The Vulnerable User
- A2.3.2 – Security by Encryption
- A2.3.3 – World Security, World Democracy
- P2.3.4 – The Heist
Cross Curricular Connections:

- ELA:
  - Reading
  - Writing
  - Speaking and listening
- Math:
  - Arithmetic
  - Algebraic expressions
  - Logic problems

Depth of Knowledge (Section 5)

DOK: 4
Curriculum: Computer Science Principles (PLTW)

Curricular Unit: Analyzing, Manipulating, and Drawing Conclusions from Data

Instructional Unit: I. Analyze and visualize data related to any discipline or regarding any question

Standard Alignments (Section 2)

<table>
<thead>
<tr>
<th>MGGLE: PS.1.C; PS.3.A,B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge: (CA) 1,3,4,6 (MA) 1,3-6</td>
</tr>
<tr>
<td>CCSS: 9-10.RI.1; 9-10.RI.2; 9-10.RI.4; 9-10.RI.6; 9-10.RI.8; 9-10.W.1a-e; 9-10.W.2a-f; 9-10.W.4; 9-10.W.5; 9-10.W.6; 9-10.W.7; 9-10.W.8; 9-10.W.9; 9-10.W.10; 9-10.SL.1a-d; 9-10.SL.2; 9-10.SL.3; 9-10.SL.4; 9-10.SL.6; 9-10.L.1a,b; 9-10.L.2c; 9-10.L.3a; 9-10.L.4a-d; 9-10.L.5b; 9-10.L.6; 9-10.RH.1; 9-10.RH.2; 9-10.RH.4; 9-10.RH.5; 9-10.RH.6; 9-10.RH.7; 9-10.RH.8; 9-10.RH.9; 9-10.RH.10; 9-10.RST.1; 9-10.RST.2; 9-10.RST.3; 9-10.RST.4; 9-10.RST.7; 9-10.RST.8; 9-10.RST.9; 9-10.RST.10; 9-10.WHST.1a,c-d; 9-10.WHST.2a-f; 9-10.WHST.4; 9-10.WHST.5; 9-10.WHST.7; 9-10.WHST.8; 9-10.WHST.9; 9-10.WHST.10; S-ID.1; S-IC.1; S-IC.2; S-IC.4; S-IC.5; S-CP.4; S-MD.4</td>
</tr>
<tr>
<td>NETS: 1a,b; 2a,c,d; 3a; 4a,b,d; 5b; 6</td>
</tr>
<tr>
<td>Performance: 1.1, 1.2, 1.4, 1.7, 1.10, 2.1-2.3, 2.5, 2.7, 3.1-3.3, 3.7, 4.5, 4.6, 4.8</td>
</tr>
</tbody>
</table>

Unit (Section 3)

Learning Targets:

- **Knowledge:**
  - Describe the role of creativity in producing data visualizations
  - Distinguish deterministic and randomized models and describe the role of Monte Carlo techniques
  - Identify data visualizations as an important tool for communication
  - Articulate a range of positions on questions related to privacy with respect to Big Data

- **Skills:**
  - Analyze a computational artifact for correctness
  - Represent and understand phenomena by identifying relationships among
  - Identify the appropriate type of basic visualization appropriate to a particular data set
  - Use appropriate tools to construct a scatter plot, histogram, pie chart, or compound bar graph
  - Use appropriate tools to create new data sets derived from other data sets
  - Use appropriate tools to describe a linear relationship (or lack thereof) inferred from data
  - Collaborate when collecting and analyzing data to answer a question
Instructional Strategies:

- The teacher will facilitate:
  - the collaboration of pair programming
  - in-class discussions over using data to make inferences
  - model different tools for manipulating and analyzing data
  - guide students in finding relevant and useful databases
- Given teacher-created scenarios, students will analyze data and make inferences

Assessments/Evaluations:

- Formative (assessed using scoring guides, observation, and practice activities) – Students will apply data collection and data visualization to an area of interest to a team:
  - Identify and describe a data set that could support a rich investigation
  - Visualize data
  - Ask insightful questions that seek connections among many significant attributes
- Summative (assessed using scoring guides, observation, and practice activities):
  - Visualize data that are provided:
    - Create visualizations
    - Analyze visualizations for clarity, correctness, and usability
    - Interact with a data visualization to explore data
  - Collaboratively transform, clean, select, and sort data that are provided:
    - Automate the manipulation of numeric and string data in a programming language
    - Automate the manipulation of numeric and string data in a spreadsheet
    - Structure and interpret the structure of data provided
    - Describe data
  - Describe the impact of digital data on society:
    - Describe societal issues raised by digital data
    - Describe how Big Data are affecting career fields and describe career opportunities related to data
    - Describe technical issues related to Big Data
    - Analyze short- and long-term beneficial and harmful impacts of a computing innovation; create a professional technical report and a creative visual artifact about the innovation

Sample Assessment Questions:

- How will computation impact fields other than computing itself?
- How will computation impact society?
- How can patterns be discovered in data?

Instructional Resources/Tools:

- A3.1.1 – Time Series and Trends
- A3.1.2 – Issues with Data
- A3.1.3 – Big Data and Parallel Processing
- A3.1.4 – Pie Charts and Bar Graphs
- A3.1.5 – Histograms and Distributions
- B3.1.6 – Datify Your Interest
Cross Curricular Connections:

- ELA:
  - Reading
  - Writing
  - Speaking and listening
- Math:
  - Arithmetic
  - Algebraic expressions
  - Logic problems

**Depth of Knowledge (Section 5)**

DOK: 4
Curriculum: Computer Science Principles (PLTW)

Curricular Unit: Analyzing, Manipulating, and Drawing Conclusions from Data

Instructional Unit: J. Manipulate datasets to compare and contrast trends and patterns to infer information

**Standard Alignments (Section 2)**

<table>
<thead>
<tr>
<th>MGGLE: PS.1.C; PS.3.A,B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge: (CA) 1,3,4,6 (MA) 1,3-6</td>
</tr>
<tr>
<td>CCSS: 9-10.W.10; 9-10.SL.1c; 9-10.SL.2; 9-10.SL.5; 9-10.SL.6; 9-10.L.1; 9-10.L.2.c; 9-10.L.3.a-d; 9-10.L.5b; 9-10.L.6; 9-10.L.7; 9-10.L.8; 9-10.RH.7; 9-10.RH.8; 9-10.RST.3; 9-10.RST.4; 9-10.RST.7; 9-10.RST.8; 9-10.RST.9; 9-10.WHST.10; S-ID.1; S-ID.2; S-ID.3; S-ID.5; S-ID.6c; S-ID.8; S-IC.1; S-IC.4; S-IC.5</td>
</tr>
<tr>
<td>NETS: 1a,b; 2a,c,d; 3a; 4a,b,d; 5b; 6</td>
</tr>
<tr>
<td>Performance: 1.1, 1.2, 1.4, 1.7, 1.10, 2.1-2.3, 2.5, 2.7, 3.1-3.3, 3.7, 4.5, 4.6, 4.8</td>
</tr>
</tbody>
</table>

**Unit (Section 3)**

- **Knowledge:**
  - Relate the categorical or quantitative nature of data to the operations and visualizations that are appropriate
  - Describe a data structure, such as a tree, and describe an algorithmic problem and multiple solutions concerning that data structure
  - Describe examples of how location-aware computing is transforming infrastructure
  - Describe examples of crowdsourcing being used for large data set collection or collective biological processing power
  - Describe examples demonstrating that the Internet has profoundly altered the way that scientists collaborate, publish, and access each other's work

- **Skills:**
  - Propose a method for simulating a natural phenomenon of interest
  - Identify quantities relevant to a phenomenon and explain the implications of abstracting to a particular model that accounts for some parameters and ignores others
  - Manipulate the parameters of a simulation to identify how the parameters affect the behavior being modeled
  - Interpret data and data visualizations to reach conclusions about a model's behavior
  - Access public information and computational resources to answer questions regarding published data
  - Use appropriate tools to represent geographic data
  - Use appropriate tools to visualize linked data
  - Use appropriate tools to detect clusters in multi-variable data
- Use appropriate tools to crowd-source data production
- Present a plan for producing knowledge

### Instructional Strategies:

- The teacher will:
  - facilitate the collaboration of pair programming
  - model different:
    - tools for manipulating and analyzing data
    - formulas and algorithms for finding and comparing data trends
  - guide students in finding relevant and useful databases
- Given teacher created situations, students will analyze data and make inferences

### Assessments/Evaluations:

- **Formative** (assessed using scoring guides, observation, and practice activities) – Students will explore and analyze data to investigate a cluster of questions on a topic of interest
  - Document an iterative process, breaking the investigation into smaller tasks
  - Explore and analyze data
  - Show awareness of scalability to larger data when choosing tools and algorithms
  - Collaborate when collecting, exploring and analyzing data and when presenting results
  - Present knowledge and visualizations
- **Summative** (assessed using scoring guides, observation, and practice activities):
  - Abstract real phenomena with a model that incorporates randomness
    - Create a model incorporating randomness that abstracts a measured phenomenon
    - Produce and analyze Monte Carlo simulation results
    - Compare differences in the proportions or means of two samples to variability implicit within the samples
    - Recognize linear correlation and lack thereof
  - Improve accuracy when informally inferring patterns (and the lack thereof) in bivariate data
  - Describe operations that can be performed with linked data: Manipulate and visualize linked data
  - Manipulate geographic data. Describe operations that can be performed on GIS data, visualize GIS data, and describe opportunities presented by GIS data:
    - Manipulate and visualize GIS data
    - Describe impact of GIS data
  - Manipulate genomic data. Describe operations that can be performed on genomic data, visualize genomic data, and describe opportunities and challenges presented by genomic data:
    - Manipulate and visualize genomic data
    - Describe opportunities and challenges of genomic data
Sample Assessment Questions:
- A3.2.1 – Inferential Statistics
- A3.2.2 – Image Data
- A3.2.3 – Linked Data
- A3.2.4 – Geographic Data
- A3.2.5 – Considering Gattaca
- P3.2.6 – Genomic Data
- B3.2.7 – Investigating with Data

Instructional Resources/Tools:
- How has computation changed biology?
- How will computation impact fields other than computing itself?
- How will computation impact society?

Cross Curricular Connections:
- ELA:
  - Reading
  - Writing
  - Speaking and listening
- Math:
  - Arithmetic
  - Algebraic expressions
  - Logic problems

**Depth of Knowledge (Section 5)**

DOK: 4
Curriculum: Computer Science Principles (PLTW)

Curricular Unit: Employing and Managing Computer Simulations and Modeling

Instructional Unit: K. Utilize modeling and simulation to understand a physical, biological, or social phenomenon

### Standard Alignments (Section 2)

| MGGLE: CD.7.A,C |
| Knowledge: (CA) 1,3,4,6 (MA) 1,3-6 |
| CCSS: 9-10.W.10; 9-10.SL.1a,c; 9-10.SL.2; 9-10.SL.6; 9-10.L.1; 9-10.L.2c; 9-10.L.4a-d; 9-10.L.5b; 9-10.L.6; 9-10.RH.2; 9-10.RST.2; 9-10.RST.3; 9-10.RST.4; 9-10.RST.6; 9-10.RST.7; 9-10.WHST.10; N-Q.2 |
| NETS: 1a,c,d; 2a,c,d; 3-6 |
| Performance: 1.4, 1.5, 1.10, 2.1, 2.5-2.7, 3.2-3.4, 3.7, 4.5, 4.6, 4.8 |

### Unit (Section 3)

#### Learning Targets:

- **Knowledge:**
  - Describe any simulation as abstracting some aspects of a model while ignoring other details
  - Describe any simulation as relying on assumptions, some of which can be parameterized, with conclusions applicable to a range of parameter
  - Describe an abstraction of hardware
  - Recognize that chaotic and periodic behavior can be exhibited by both deterministic and randomized models
  - Discrete, continuous, and agent-based models using digital computation rely on discrete calculations
  - Describe some of the major applications of simulation
  - Identify landmarks in hardware development such as tubes, transistors, and VLSI, as well as developments such as optoelectronics, nanotechnology, and quantum computing
  - Describe the abstraction of a logic gate and explain how complex logic functions can be constructed from NAND gates
  - Describe examples of crowdsourcing being used for large data set collection or collective biological processing power

- **Skills:**
  - Identify and explain some of the assumptions made by a particular model
  - Compare hardware and software manipulations that produce similar outcomes
  - Construct a circuit including discrete components and semiconductor chips from a symbolic diagram
Instructional Strategies:

- The teacher will:
  - facilitate:
    - the collaboration of pair programming
    - discussions on the usefulness and efficiency of using computers to simulate real-world events
  - model software used to create simulations
  - Given teacher-created simulations, students will manipulate the data to achieve desired results

Assessments/Evaluations:

- Formative (assessed using scoring guides, observation, and practice activities) – Students will:
  - design and implement a problem's solution close to the semiconductor level to produce output based upon processed input:
    - Create a meaningful circuit from discrete electronic components and explain the nature of its binary digital behavior
    - Relate the hardware solution to an abstraction independent of hardware particulars
  - explore the simulation's parameter space and draw conclusions from the data produced when provided a model and simulation:
    - Identify patterns in transient, steady state, oscillating, and/or chaotic behavior
    - Explore a model's behavior across its parameter space and identify regions of parameter space with distinct behavior
    - Identify the details parameterized in a particular model, identify some details which were abstracted away, and explain the possible implications of these decisions

- Summative (assessed using scoring guides, observation, and practice activities):
  - Describe how software interfaces with hardware – Students will relate:
    - transistors
    - integrated circuits
    - gates
    - assembly language instructions
  - Analyze short- and long-term beneficial and harmful impacts of a computing innovation
  - Create a professional technical report on the impacts of a computing innovation
  - Present a creative visual artifact about the beneficial or harmful effects of a computing innovation
  - Analyze the impact of some innovation involving:
    - distributed computing
    - parallel computing
    - high performance computing
    - modeling and simulation
    - crowd sourcing
  - Interpret the results of a simulation from a visualization: Interpret a time-series visualization of simulation results
Sample Assessment Questions:

- How are simulations created from models?
- How are simulation models similar to and different from reality?
- How are modeling and simulation impacting other creative fields?

Instructional Resources/Tools:

- A4.1.1 – Computing Impacts All Fields
- A4.1.2 – Basic Control Circuits
- A4.1.3 – Introducing Simulations
- A4.1.4 – Varying Parameters
- A4.1.5 – Assumptions, Abstractions, and Ethics

Cross Curricular Connections:

- ELA:
  - Reading
  - Writing
  - Speaking and listening
- Math:
  - Arithmetic
  - Algebraic expressions
  - Logic problems

**Depth of Knowledge (Section 5)**

DOK: 4
Curriculum: Computer Science Principles (PLTW)

Curricular Unit: Employing and Managing Computer Simulations and Modeling

Instructional Unit: L.
- Understand and communicate about real systems by abstracting and simulating them and interpreting the results
- Contrast the assumptions and predictive power of simulations

Standard Alignments (Section 2)

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge:</td>
<td>(CA) 1,3,4,6 (MA) 1,3-6</td>
</tr>
<tr>
<td>NETS: 1a,c,d; 2a,c,d; 3-6</td>
<td></td>
</tr>
<tr>
<td>Performance: 1.4, 1.5, 1.10, 2.1, 2.5-2.7, 3.2-3.4, 3.7, 4.4-4.6, 4.8</td>
<td></td>
</tr>
</tbody>
</table>

Unit (Section 3)

Learning Targets:
- Knowledge:
  - Give examples of synergetic properties that can emerge from real or simulated collections of independent agents

Instructional Strategies:
- The teacher will:
  - facilitate the collaboration of pair programming
  - model and guide students through using algorithms in real world simulations
  - provide real-world simulations for students to manipulate and analyze
  - lead pair discussions about drawing conclusions from simulations
  - guide students through the process of using computer simulations to predict real-world events

Assessments/Evaluations:
- Formative (assessed using scoring guides, observation, and practice activities) – Students will present the abstractions, algorithms, and mathematical concepts used by a development team to strategically satisfy a purpose:
  - Explain the use of algorithms and abstraction in a program
  - Present a development process by which a program was collaboratively developed for a purpose
- Use mathematics and logic concepts, abstraction, and algorithms creatively and effectively
- Create readable code
- Summative (assessed using scoring guides, observation, and practice activities):
  - Envision positive and negative impacts of a program: Describe how a program can impact people
  - Give examples of emergent phenomena: Identify emergence of phenomena in agent-based modeling and distributed computing
  - Analyze the results of a simulation and describe how the discrete nature of digital data affects simulation:
    - Interpret the results of a simulation
    - Describe the algorithm of a simulation in which continuous time and space are modeled with a finite number of discrete intervals
    - Given a model for a phenomenon with fractal or chaotic behavior, relate the validity of the simulation results to the precision of the model's mesh, parameters, and initial conditions
    - Explain why high performance computing can extend the validity of modeling and simulation

Sample Assessment Questions:
- How has computation affected our ability to predict the future?
- How has computation affected our ability to experience virtual phenomena?
- How has simulation changed the design process in engineering and other creative fields?

Instructional Resources/Tools:
- A4.2.1 – Emergent Behavior
- A4.2.2 – Neural Networks
- P4.2.3 – Modifying a Simulations Assumptions
- A4.2.4 – Beauty In Chaos And Fractals
- P4.2.5 – Computer Science Principles

Cross Curricular Connections:
- ELA:
  - Reading
  - Writing
  - Speaking and listening
- Math:
  - Arithmetic
  - Algebraic expressions
  - Logic problems

**Depth of Knowledge (Section 5)**

DOK: 4