CONTENT

MIDDLE SCHOOL CODING

BOARD APPROVAL DATE: 8/17/21

BOARD ADOPTION OF STATE STANDARDS: 9/1/2022

Unit Overview (Standards Coverage)				
Unit	**Standards**	**Unit Focus**	**Skills Overview**	**Suggested Pacing**
Unit 1	**8.1.8.DA.5, 8.1.8.CS.4, 8.1.8.AP.1, 8.1.8.AP.2, 8.1.8.AP.3, 8.1.8.AP.4, 8.1.8.AP.5, 8.1.8.AP.6, 8.1.8.AP.7, 8.1.8.AP.8,8.1.8.AP.9**	**Block Programming**	**Create algorithms, decompose a problem, collect, design, test & execute a program**	**One marking period/every other day (less 4 days)**
Unit 2	**8.1.8.CS.1, 8.1.8.CS.2, 8.1.8.CS.4, 8.1.8.IC.1, 8.1.8.AP.7, 8.1.8.AP.8,8.1.8.AP.9**	**App Development**	**Create algorithms, decompose a problem, collect, analyze and represent data, design, test & execute a program.**	**Half a marking period/ every other day**
Unit 3	**8.1.8.IC.2, 8.1.8.AP.4, 8.1.8.AP.7, 8.1.8.AP.8, 8.1.8.AP.9**	**Web Design**	**Students create simple HTML pages.**	**Half a marking period/every other day**
Unit 4	**8.1.8.CS.1, 8.1.8.CS.2, 8.1.8.CS.3, 8.1.8.CS.4, 8.1.8.NI.1, 8.1.8.NI.2, 8.1.8.NI.3, 8.1.8.NI.4**	**Network & Internet**	**Troubleshooting computer problems, cybersafety**	**4 days**

This document outlines in detail the answers to following four questions:

1. **What do we want our students to know?**
2. **How do we know if they learned it?**
3. **What do we do if they did not learn it?**
4. **What do we do when they did learn it?**

Unit 1 Block Programming		
Content & Practice Standards (write in full)	**Suggested Standards for Practice**	**Critical Knowledge & Skills**
• **8.1.8.CS.4: Systematically apply troubleshooting strategies to identify and resolve hardware and software problems in computing systems.** • **8.1.8.DA.5: Test, analyze, and refine computational models** • **8.1.8.AP.1: Design and illustrate algorithms that solve complex problems using flowcharts and/or pseudocode.** • **8.1.8.AP.2: Create clearly named variables that represent different data types and perform operations on their values.** • **8.1.8.AP.3: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.** • **8.1.8.AP.4: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.** • **8.1.8.AP.5: Create procedures with parameters to organize code and make it easier to reuse.** • **8.1.8.AP.6: Refine a solution that meets users' needs by incorporating feedback from team members and users.** • **8.1.8.AP.7: Design programs, incorporating existing code, media, and libraries, and give attribution.** • **8.1.8.AP.8: Systematically test and refine programs using a range of test cases and users** • **8.1.8.AP.9: Document programs in order to make them easier to follow, test, and debug.**	• **CRP6. Demonstrate creativity and innovation** • **CRP8. Utilize critical thinking to make sense of problems and perseverance in solving them.** • **9.2.8.CAP.12: Assess personal strengths, talents, values, and interests to appropriate jobs and careers to maximize career potential.**	• **Computational thinking** • **Computer Programming** • **Creative development**

Unit 1 Block Programming	
Stage 1 – Desired Results	
Unit Summary	**Core and Supplemental Materials/Resources (open resources)**
Students will program their own interactive stories, games, and animations and share these creations with others in the studio. Block programming teaches students to think creatively and reason systematically,	https://scratch.mit.edu/ https://sites.google.com/a/sfusd.edu/mycs-teacher/quarter www.code.org videos for concepts and careers
Understandings	
Students will understand that computer programs are a sequence of instructions. Students will understand how to write programs using block-based programming languages.	

Students will understand algorithms in relation to coding.
Students will understand how to design, code, test and execute a program in Scratch.
Students will understand how to decompose a problem into smaller, more manageable parts.
Students will understand how to locate and debug errors in a program.

Students will know…	**Students will be able to…**
Students will know how to use the Scratch interface. *Students will know how to create an animation, story or game using block coding.* *Students will know what the following terms mean: program, event, condition, algorithm, loop, bug, and variable.*	*Write programs using block based programming languages.* *Locate and debug errors in their program.* *Identify which commands create the effects they want.* *Understand and apply looping, conditional and events.* *Broadcast messages* *Understand and use variables to store and display data.*

Stage 2 – Assessment Evidence

Performance Tasks:	Other Evidence **(Alternate Assessments)**:
Scratch Activities *Animate a Name* *Race to the Finish* *Shark Attack* *Original game, animation or story* *Code.org - Accelerated Computer Science course*	*Teacher observation* *Participation*

Stage 3 – Learning Plan

Student's performance obligations:

- Students will gain foundational knowledge of basic concepts such as, conditionals, loops, events and variables to help prepare them for a culminating assignment. The student's final performance task will be to have students create an original game, story or animation that includes these concepts. Student work will be judged for understanding by following rubrics for assessment on activities. Students are also graded on their progress in code.org.

Hooking the students:

- The hook for students will be through computer science themed videos introducing the subject matter. These videos show how technology is changing everything, career videos (example digital lighting movies with Danielle Feinstein from Pixar, digital costumer designer, Fran Kalel, diversity in computer science.
- Completed exemplary projects are shown to motivate students on what they are able to create.

Explore and Equip. 21st Century Learning and Interdisciplinary connections.

- Students code stories, animations and games in Scratch.
- By learning on code.org students learn basic computer science concepts in a fun, engaging way. They create computer programs including interactive games and creative projects

Organize and sequence learning:

- Students begin by tinkering in Scratch so they can gain familiarity with the Scratch interface and learn how to share in the class studio.
- Students animate their name in code so they can learn the events, sound, motion and look blocks.
- To learn the relationship of the canvas to the coordinate plane and sequencing, students code a Space Exploration project.
- Students code a Race to the Finish game to learn loops and conditional statements.
- The next project is the shark game to learn how to code variables (score). In addition, the game utilizes conditional blocks to re-inforce what they learned in the Race to the Finish game.
- The final culminating project allows students to create an original game, story or animation that includes all the learned concepts.

Pre-assessments:

- We are able to assess the student's coding ability from the first project which animates the letters in their name.
- Students will be asked to write pseudocode for some of their projects.
- The final project requires students to brainstorm ideas before coding their event.

Acquisition, meaning, and transfer:

- Instead of just using technology, students will learn to become creators of technology. Students acquire the skills, learn the concepts and transfer their knowledge into coding the animations, stories and games in Scratch.

Reflect principles of learning and best practices in the learning plan:

- As we refine our coding program we can adapt, edit, and learn from the successes and missteps along their way. Each new year brings with it the opportunity to deepen our practice and refine the students' learning experiences with Scratch and code.org.
- Experiential. Active hands on in Scratch.
- Authentic. Students use computer concepts to create their animation, game or story.
- Challenging. The rigor rises as students progress through the activities.

Engaging students:

- Students are very engaged in creating the animations, games and stories. Not every student loves coding, but they are given enough freedom to create something that appeals to them. If someone prefers art, they will do an animation that is very visual. If they prefer video games, they will choose to create game -like projects.

PROGRESS MONITORING

Monitor students' progress toward acquisition, meaning-making, and transfer, during lesson events:

- Students' work is shared in the class studios for assessment. From the shared projects, we are able to identify who is struggling with the computer concepts and needs additional help.
- Code.org has a teacher dashboard that contains a variety of tools for checking on and evaluating student work, as well as managing the students in a particular section.

Potential rough spots and student misunderstandings:

- Students can become frustrated when their program does not work correctly.
- Students need to learn to debug their own programs

Student feedback:

- Teacher will circulate the room.
- Teacher will provide frequent feedback and answer student questions.
- In code.org, students are given hints and videos for further explanation before the step is autograded.

Student support:

- Reteaching, one-on-one teacher conference, modified assignments

Planned Differentiation & Interventions for Tiers I, II, III, ELL, 504s, SPED, and Gift & Talented Students

• Rethink and revise. Dig deeper into ideas at issue (through the faces of understanding). Revise, rehearse, and refine, as needed. Guide students in self-assessment and self-adjustment, based on feedback from inquiry, results, and discussion.

• Evaluate understanding:. Reveal what has been understood through final performances and products. Involve students in a final self-assessment to identify remaining questions, set future goals, and point toward new units and lessons.

•Tailor (personalize) the work to ensure maximum interest and achievement. Differentiate the approaches used and provide sufficient options and variety (without compromising goals) to make it most likely that all students will be engaged and effective.

Gifted & Talented:

Continue with code.org Accelerated course after completing Scratch activities. When completed code.org students go to Intro to Java - Khan Academy

Tier I: Continue with code.org Accelerated course after completing Scratch activities.
Tier II: Students will receive some additional support through teacher conferencing to aid in completing activities. Students will have additional teacher support in understanding the code and will be retaught as necessary.
Tier III: Students will receive a lot of additional support through teacher conferencing to aid in completing activities. Students will have additional teacher support in understanding the code and will be retaught as necessary. If necessary, the activity requirements will be cut for students.
ELL: Students have access to Google translate. Scratch is offered in 70 languages. Depending on English literacy proficiency, students may have assignments and assessments printed in their native language. Students may have extra time to work on activities with ELL teachers.
504s: Accommodations will be provided according to the 504 plan. Example accommodations: preferential seating, extra time to complete assignments, extra support, breaks, etc.
SPED: Follow student's IEP. Example accommodations: preferential seating, extra time to complete assignments, extra support, breaks, etc. Ensure that students understand the directions and know how to complete the task.

Unit 2 App Development		
Content & Practice Standards	**Suggested Standards for ELA Practice**	**Critical Knowledge & Skills**
• 8.1.8.CS.4: Systematically apply troubleshooting strategies to identify and resolve hardware and software problems in computing systems. • 8.1.8.DA.5: Test, analyze, and refine computational models • 8.1.8.AP.1: Design and illustrate algorithms that solve complex problems using flowcharts and/or pseudocode. • 8.1.8.AP.2: Create clearly named variables that represent different data types and perform operations on their values. • 8.1.8.AP.3: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. • 8.1.8.AP.4: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. • 8.1.8.AP.5: Create procedures with parameters to organize code and make it easier to reuse. • 8.1.8.AP.6: Refine a solution that meets users' needs by incorporating feedback from team members and users. • 8.1.8.AP.7: Design programs, incorporating existing code, media, and libraries, and give attribution. • 8.1.8.AP.8: Systematically test and refine programs using a range of test cases and users • 8.1.8.AP.9: Document programs in order to make them easier to follow, test, and debug.	• CRP6. Demonstrate creativity and innovation • CRP8. Utilize critical thinking to make sense of problems and perseverance in solving them. • 9.2.8.CAP.12: Assess personal strengths, talents, values, and interests to appropriate jobs and careers to maximize career potential.	• Computational thinking • Computer Programming

Unit 2 App Development/Coding	
Stage 1 – Desired Results	
Unit Summary	**Core and Supplemental Materials/Resources (open resources)**
Students will create an app based on a specification using block code, design the user interface, run the program and debug anything not working properly.	https://appinventor.mit.edu/ or https://code.org/educate/applab https://sites.google.com/site/middleschoolandroid/home/app-inventor-tutorials/day-1 https://docs.google.com/document/d/1Wf9oUuRi5iNb2cVrn1F8kbDDys0CcAcN-m4ZLpXohmY/edit
Understandings	
Students will understand that computer programs are a sequence of instructions. Students will understand that you can write programs using block-based programming languages. Students will understand that there are different methods of input or output in common apps. Students will understand that an app is designed in one view and then programmed in a different view.	
Students will know…	**Students will be able to…**
Students will know how to use the App Inventor interface. *Students will know how to use block-style programming.* *Students will know how to design a user interface using layout elements.* *Students will identify different methods of input or output in common apps.* *Students will know how to design, develop and publish their own mobile app.* *Students will know how to tell the difference between design view and code view.* *Student will know how to change the properties of the components*	*Decompose a problem, into smaller, more manageable parts.* *Write programs using visual (block-based) programming languages.* *Write programs using if and if/else blocks.* *Write programs that display images and play sounds.* *Create algorithms, or series of ordered steps, to solve problems.* *Identify careers that utilize computing and technology.*
Stage 2 – Assessment Evidence	
Performance Tasks: *Students will complete the following activities:* *Day 1 : Bouncing Balls / Variables* *Day 2 : Block Editor / Buttons and Labels* *Day 3 : Calculator / If Statements* *Day 4 : HelloPurr / Media* *Day 5 : PaintPot / Events* *Day 6: I Have a Dream/If Statements, Media*	*Other Evidence* **(Alternate Assessments)**: Teacher observation Participation

Students demonstrate their understanding: ● *Students will demonstrate understanding by designing and coding each app listed above.* ● *Students will demonstrate dispositions amenable to open-ended problem solving and programming.*	

Stage 3 – Learning Plan

Student's performance obligations:

- The activities are headed towards an increased understanding of user interface, conditionals, variables, media and events. Students will be comfortable knowing how to test, locate errors, and debug their own programs.

Hooking the students:

- Completed exemplary projects are shown to motivate students on what they are able to create.

Explore and Equip. 21st Century Learning and Interdisciplinary connections:

- The student will act on creative ideas to make a tangible and useful contribution to the field in which innovation will occur.
- Elaborate, refine, analyze and evaluate their own ideas in order to improve and maximize creative efforts.
- Solve different kinds of non-familiar problems in both conventional and innovative ways.

Organization and sequence learning

- Students learn the app inventor interface through building a simple app-Paint Pot.
- The Bouncing Balls gives students an understanding of how variables are used in coding apps.
- The block editor lesson continues modifying the Bouncing ball app adding buttons and labels.
- The Calculator activity introduces conditional blocks.
- The Hello Purr activity has students add sound to their app.
- The I Have a Dream activity uses both conditionals and media files to transition between the two famous speeches.

Pre-assessments:

- We are able to assess the student's ability from the block coding unit. The apps are also created with block coding.

Acquisition, meaning, and transfer:

- Students acquire knowledge after completing the hour of code app activity.
- Students transfer this knowledge to build an app from scratch.

Principles of learning and best practices:

- Experiential. Active hands on activities
- Authentic. MIT AppLab can be used by professional web designers.
- Challenging. Almost all students are new to app design and development.

Engagement:

- While students make apps, they learn valuable STEAM skills like coding, graphic design, content creation, and how to solve real-world problems with digital design. Students can see real life learning and learning how to build an app inspires intellectual and creative empowerment

PROGRESS MONITORING

Monitor students' progress toward acquisition, meaning-making, and transfer, during lesson events:

- Teacher will circulate while students are following tutorials in MIT App Inventor and working in the App Inventor interface.
- Students will first watch the teacher model the app process, then work independently to complete the task.

Rough spots and student misunderstandings:

- Students may have difficulty following the video tutorial.
- Students can become frustrated when their program does not work correctly.
- Students need to learn to debug their own programs

Student feedback:

- Teacher will circulate the room.
- Teacher will provide frequent feedback and answer student questions.

Student support:

- Reteaching, one-on-one teacher conference, modified assignments

Planned Differentiation & Interventions for Tiers I, II, III, ELL, 504s, SPED, and Gift & Talented Students

• Rethink and revise. Dig deeper into ideas at issue (through the faces of understanding). Revise, rehearse, and refine, as needed. Guide students in self-assessment and self-adjustment, based on feedback from inquiry, results, and discussion.

• Evaluate understandings. Reveal what has been understood through final performances and products. Involve students in a final self-assessment to identify remaining questions, set future goals, and point toward new units and lessons.

•Tailor (personalize) the work to ensure maximum interest and achievement. Differentiate the approaches used and provide sufficient options and variety (without compromising goals) to make it most likely that all students will be engaged and effective.

Gifted & Talented:
Students who complete MIT App Inventor are assigned to Khan Academy Intro to Java - Khan Academ or they will have an option to create another app following a video tutorial. (drum app)

Tier I:
Students who finish an exercise early will be given video instructions to design a drum app.

Tier II:
Students will receive some additional support through teacher conferencing to aid in further understanding and completion of the activity. Constant teacher monitoring will determine what students need to be retaught if necessary.

Tier III:
Students will receive a lot of additional support to help them understand the MIT App interface and coding blocks . Students will have additional teacher support such as a breakdown of the block code needed. Some students may not be required to have all the components as assigned.

ELL: Students have access to Google translate. MIT App Inventor is offered in Spanish & Chinese. Depending on English literacy proficiency, students may have assignments and assessments printed in their native language. Students may have extra time to work on activities with ELL teachers.
504s: Accommodations will be provided according to the 504 plan. Example accommodations: preferential seating, extra time to complete assignments, extra support, breaks, etc.
SPED: Follow student's IEP. Example accommodations: preferential seating, extra time to complete assignments, extra support, breaks, etc. Ensure that students understand the directions and know how to complete the task.

Unit 3 WEB DESIGN/CODING		
Content & Practice Standards	**Suggested Standards for ELA Practice**	**Critical Knowledge & Skills**
• **8.1.8.IC.2: Describe issues of bias and accessibility in the design of existing technologies.** • **8.1.8.AP.4: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.** • **8.1.8.AP.7: Design programs, incorporating existing code, media, and libraries, and give attribution.** • **• 8.1.8.AP.8: Systematically test and refine programs using a range of test cases and users.** • **8.1.8.AP.9: Document programs in order to make them easier to follow, test, and debug**	• **CRP6. Demonstrate creativity and innovation** • **CRP8. Utilize critical thinking to make sense of problems and perseverance in solving them.** • **9.2.8.CAP.12: Assess personal strengths, talents, values, and interests to appropriate jobs and careers to maximize career potential.**	• **Computational thinking** • **Web development** • **Creative thinking**

Unit 3 WEB DESIGN/CODING	
Stage 1 – Desired Results	
Unit Summary	**Core and Supplemental Materials/Resources (open resources)**
In this unit, students create their own web pages and display them in their web browser. Students learn the tags to build their webpage using Hyper Text Markup Language (HTML)	Notepad Web browser https://curriculum.code.org/csd-20/unit2/ https://codehs.com/
Understandings	
Students will understand that web pages can be used as a medium for creative expression. Students will understand how the web works Students will understand how markup language is used to create web pages. Students will understand they need to think more critically about the impact of sharing information online and how to be more critical consumers of content.	
Students will know…	**Students will be able to…**
Students will know the structure of a webpage. *Students will learn the language HTML.* *Students will know the basic tags required for all HTML documents.* *Students will know how to create a blank HTML page with all of the essential tags in place.*	*Students will be able to create a simple text program in Notepad using basic HTML tags.* *Students will be able to create a web page which will include the following elements: an image, links, an ordered list, and an unordered list* *Students will be able to find and fix their mistakes.*

	Students will be able to consider the impact of the choices they make when creating and consuming digital content
Stage 2 – Assessment Evidence	
Performance Tasks: *Introduction to web development in code.hs.com* *Create a SimpleText program in Notepad using basic HTML tags.* *Students create a simple web page using more HTML formatting tags and an image.* *Students create a simple web page including a list.* *Students create a simple web page including an unordered list.* *Create their own personal page which combines all of the tags they have learned in the unit.*	Other Evidence: *Teacher observation* *Participation*
Stage 3 – Learning Plan	

Student's performance obligations:

- Students start on codehs.com to get an introduction to HTML, then progress to the structure of HTML, which leads into formatting in HTML. Learning the tags and structure enables students to create their own web page including an image, hyperlinks and lists.

Hooking the students:

- Students are hooked after having them go into the Inspector of any webpage they choose, and looking at the source code. Students are intrigued they are able to see code and are surprised at how much goes into the makeup of a web page.

Explore and Equip. 21st Century Learning and Interdisciplinary connections:

- Students create simple web pages in Notebook and then add more HTML tags as they progress.
- Students are introduced to web computer science careers.

Organization and sequence learning

- Create a SimpleText program on Notepad using basic HTML tags.
- Students create a simple web page using more HTML formatting tags and an image.
- Students create a simple web page including a list.
- Students create a simple web page including an unordered list.
- Students create their own personal web page.

Pre-assessments

- Ask students three things they know about HTML prior to learning it.
- The final web page assignment requires students to brainstorm ideas before starting to code.

PROGRESS MONITORING
Monitor students' progress toward acquisition, meaning-making, and transfer, during lesson events:

- Students will view slides that show how each of the HTML tags are used for the day's lesson.
- Teacher will circulate while students are creating their pages.

Potential rough spots and student misunderstandings:

- Students can become frustrated when their web page does not look or work properly.
- Students need to learn to debug their own programs

Student feedback:

- Teacher will circulate the room.
- Teacher will provide frequent feedback and answer student questions.
- In code.org, students are given hints and videos for further explanation before the step is autograded.

Student support:

- Reteaching, one-on-one teacher conference, modified assignments

Planned Differentiation & Interventions for Tiers I, II, III, ELL, 504s, SPED, and Gift & Talented Students

• Rethink and revise. Dig deeper into ideas at issue (through the faces of understanding). Revise, rehearse, and refine, as needed. Guide students in self-assessment and self-adjustment, based on feedback from inquiry, results, and discussion.

• Evaluate understandings. Reveal what has been understood through final performances and products. Involve students in a final self-assessment to identify remaining questions, set future goals, and point toward new units and lessons.

•Tailor (personalize) the work to ensure maximum interest and achievement. Differentiate the approaches used and provide sufficient options and variety (without compromising goals) to make it most likely that all students will be engaged and effective.

Gifted & Talented:
Students are provided with a list of additional tags. Students add additional elements to their web page. (marquee, nested lists, formatting, etc.)

Tier I:
Students are provided with a list of additional tags. Students add additional elements to their web page. (marquee, nested lists, formatting, etc.)

Tier II:
Students will receive some additional support through teacher conferencing to aid in further understanding and completion of the activity. Constant teacher monitoring will determine what students need to be retaught if necessary.

Tier III: Students will receive a lot of additional support to help them understand the HTML tags and how to set them up in the Notepad document. Some students may not be required to have all the components as assigned.
ELL: Students have access to Google translate. Depending on English literacy proficiency, students may have assignments and assessments printed in their native language. Students may have extra time to work on activities with ELL teachers.
504s: Accommodations will be provided according to the 504 plan. Example accommodations: preferential seating, extra time to complete assignments, extra support, breaks, etc.
SPED: Follow student's IEP. Example accommodations: preferential seating, extra time to complete assignments, extra support, breaks, etc. Ensure that students understand the directions and know how to complete the task.

Unit 4 7th & 8th Networks & the Internet		
Content & Practice Standards	**Suggested Standards for ELA Practice**	**Critical Knowledge & Skills**
• **8.1.8.CS.1: Recommend improvements to computing devices in order to improve the ways users interact with the devices.** • **8.1.8.CS.2: Design a system that combines hardware and software components to process data.** • **8.1.8.CS.3: Justify design decisions and explain potential system trade-offs** • **8.1.8.CS.4: Systematically apply troubleshooting strategies to identify and resolve hardware and software problems in computing systems.** • **8.1.8.NI.1: Model how information is broken down into smaller pieces, transmitted as addressed packets through multiple devices over networks and the Internet, and reassembled at the destination.** • **8.1.8.NI.2: Model the role of protocols in transmitting data across networks and the Internet and how they enable secure and errorless communication.** • **•8.1.8.NI.3: Explain how network security depends on a combination of hardware, software, and practices that control access to data and systems.** • **8.1.8.NI.4: Explain how new security measures have been created in response to key malware events.**	• **CRP2 Apply appropriate academic and technical skills.** • **CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.**	• **Design thinking** • **Troubleshooting computer problems**

Unit 4 Networks & the Internet	
Stage 1 – Desired Results	
Unit Summary	**Core and Supplemental Materials/Resources (open resources)**
Overview on what a computer is with an explanation of hardware vs. software. Students understand computer networks process and share data. Learn troubleshooting strategies. Students compare two devices and make design changes to their chosen device. Students learn how to keep their information more secure.	https://code.org/ https://www.khanacademy.org/computing (videos)

UNDERSTANDINGS	
Students will understand that software and hardware determine a computing system's capability to store and process information. Students will understand protocols, packets and addressing are the key components for reliable delivery of information across networks. Students will understand the evolution of malware leads to understanding the key security measures and best practices needed to proactively address the threat to digital data. Students will understand it is important to take a structured step-by-step approach to effectively troubleshoot problems.	
Students will know…	**Students will be able to…**
Computing devices typically do not operate in isolation. *Networks connect computing devices to share information and resources and are an increasingly integral part of computing.* *Networks and communication systems provide greater connectivity in the computing world.* *Students will know how to apply strategies for identifying and solving routine problems that occur during everyday computer use.* *Cybersecurity is an important field of study and there is a need to protect sensitive data.* *Students understand components of computer software & hardware.*	*Describe the components and functions of computer systems and networks.* *Design modifications to computing systems in order to improve the ways users interact with the devices.* *Systematically apply troubleshooting strategies to identify and resolve hardware and software problems.* *Explain potential security threats and security measures to mitigate threats.* *Understand what malware is and have some techniques to avoid it*
Stage 2 – Assessment Evidence	
Performance Tasks: ● *Students defeat cyber attacks in an online Cybersecurity Lab game setting by defending a company that is the target of increasingly sophisticated cyber attacks. Students crack passwords, craft code, and defeat malicious hackers..* ● *Students need to research and compare a Dell computer and a macbook. Students make a pretend purchase for one of them and make a modification to improve the user experience.* ● *Students learn troubleshooting strategies through a pear deck activity.* ● *Students learn how to keep their private information more secure by checking the security of various passwords.*	Other Evidence **(Alternate Assessments)**: *Teacher observation* *Participation* *Group discussion*

Stage 3 – Learning Plan

Student's performance obligations:

- Students complete the Purchase and Improvement Computing worksheet, troubleshooting peardeck, play a cybersecurity Nova lab game, use the Password checker to determine the security of passwords and view a video on computer networking through Khan Academy.

Explore and Equip. 21st Century Learning and Interdisciplinary connections:

- Students learn basic strategies to fix their computers in their own home.
- Students have an understanding of how computers work which could encourage them to investigate careers in this field.
- Students learn how to keep their own information secure from cyber attacks.

Organization and sequence learning*:*

- Students start with an exercise where they research/compare/choose a computing device then make modifications to improve the ways users interact with it.
- Students then complete a troubleshooting activity in pear deck.
- Students learn about keeping information safe by playing The Cybersecurity Lab game, designed to teach people how to keep their digital lives safe, spot cyber scams, learn the basics of coding, and defend against cyber attacks.
- Students learn the basics of computer networking through a Khan Academy video.

Acquisition, meaning and transfer:

- Students learn skills that they can use in their own lives such as staying safe on the Internet and troubleshooting computer problems.

Pre-assessments:

- Ask students what makes a secure password.

PROGRESS MONITORING

Monitor students' progress toward acquisition, meaning-making, and transfer, during lesson events:

- Students may not understand what the terminology is for computer troubleshooting.
- Students may have difficulty understanding how the cybersecurity game works.

Student feedback:

- Teacher will circulate the room.
- Teacher will provide frequent feedback and answer student questions.

Student support:

- Reteaching, one-on-one teacher conference, modified assignments

Planned Differentiation & Interventions for Tiers I, II, III, ELL, 504s, SPED, and Gift & Talented Students
• Rethink and revise. Dig deeper into ideas at issue (through the faces of understanding). Revise, rehearse, and refine, as needed. Guide students in self-assessment and self-adjustment, based on feedback from inquiry, results, and discussion. *• Evaluate understandings. Reveal what has been understood through final performances and products. Involve students in a final self-assessment to identify remaining questions, set future goals, and point toward new units and lessons.* *•Tailor (personalize) the work to ensure maximum interest and achievement. Differentiate the approaches used and provide sufficient options and variety (without compromising goals) to make it most likely that all students will be engaged and effective.*
Gifted & Talented: If students complete the activities early, they will go to the assigned code.org online course.
Tier I: If students complete the activities early, they will go to the assigned code.org online course.
Tier II: Students will receive some additional support through teacher conferencing to aid in further understanding and completion of the activities. Constant teacher monitoring will determine what students need to be retaught if necessary.
Tier III: Students will receive a lot of additional support to help them understand the activities. Some students may not be required to complete everything as assigned.
ELL: Students have access to Google translate. Depending on English literacy proficiency, students may have assignments and assessments printed in their native language. Students may have extra time to work on activities with ELL teachers.
504s: Accommodations will be provided according to the 504 plan. Example accommodations: preferential seating, extra time to complete assignments, extra support, breaks, etc.
SPED: Follow student's IEP. Example accommodations: preferential seating, extra time to complete assignments, extra support, breaks, etc. Ensure that students understand the directions and know how to complete the task.