2025 Cyber Foundations I

Program CIP: 11.0701 — Computer Science

Direct inquiries to:

Project Manager
Research and Curriculum Unit
P.O. Drawer DX
Mississippi State, MS 39762
662.325.2510
helpdesk@rcu.msstate.edu

Program Supervisor
Office of Career and Technical Education
Mississippi Department of Education
P.O. Box 771
Jackson, MS 39205
601.359.3974

Published by:

Office of Career and Technical Education
Mississippi Department of Education
Jackson, MS 39205

Research and Curriculum Unit
Mississippi State University
Mississippi State, MS 39762

The Research and Curriculum Unit (RCU), located in Starkville, as part of Mississippi State University (MSU), was established to foster educational enhancements and innovations. In keeping with the land-grant mission of MSU, the RCU is dedicated to improving the quality of life for Mississippians. The RCU enhances the intellectual and professional development of Mississippi students and educators while applying knowledge and educational research to the lives of the people of the state. The RCU works within the contexts of curriculum development and revision, research, assessment, professional development, and industrial training.

Table of Contents

Acknowledgments

The Cyber Foundations I curriculum was presented to the Mississippi State Board of Education on January 16, 2025. The following persons were serving on the state board at the time:

Dr. Lance Evans, State Superintendent of Education, Executive Secretary
Mr. Glen V. East, Chair
Mr. Matt Miller, Vice-Chair
Dr. Ronnie L. McGehee
Mr. Bill Jacobs
Mr. Mike Pruitt
Ms. Mary Werner
Dr. Wendi Barrett
Ms. Billye Jean Stroud
Mr. Matt Mayo
Ms. Kate Riddle, Student Representative
Mr. Crosby Parker, Student Representative

The following Mississippi Department of Education (MDE) and RCU managers and specialists assisted in the development of the Cyber Foundations I:

Brett Robinson, the associate state superintendent of the MDE Office of Career and Technical Education and Workforce Development, supported the RCU and teachers throughout the development of the framework and supporting materials.
Dr. Louella Mack-Webster, the Computer Science program supervisor of the MDE Office of CTE, supported the RCU and teachers throughout the development of the framework and supporting materials.
Betsey Smith, the director of the RCU, supported RCU staff and teachers throughout the development of this framework and supporting materials.
Courtney McCubbins, the curriculum and assessment manager of the RCU, supported RCU staff and teachers throughout the development of this framework and supporting materials.
Kyle McDill, a project manager with the RCU, researched and co-authored this framework.

Special thanks are extended to the educators who contributed to the development and revision of this framework and supporting materials:

Mary Dunaway, Rankin County Schools, Brandon
Ann Thomas, Center for Cyber Education, Starkville
Anthony Emmons, Lamar County Schools, Hattiesburg
Rhonda Murph-Johnson, Jackson Public Schools, Jackson

Appreciation is expressed to the following professionals who provided guidance and insight throughout the development process:

Shelly Hollis, Center for Cyber Education, Starkville
EJ Presley, Transfer VR, Oxford
Amanda Taylor, Center for Cyber Education, Starkville

Standards

Standards and alignment crosswalks are referenced in the appendices. Depending on the curriculum, these crosswalks should identify alignment to the standards mentioned below, as well as possible related academic topics as required in the Subject Area Testing Program in Algebra I, Biology I, English II, and U.S. History from 1877, which could be integrated into the content of the units. Mississippi's CTE Cyber Foundations I is aligned to the following standards:

2018 Mississippi College- and Career-Readiness Standards for Computer Science
In an effort to closely align instruction for students who are progressing toward postsecondary study and the workforce, the *2018 Mississippi College- and Career-Readiness Standards (MS CCRS) for Computer Science* includes grade- and course-specific standards for K-12 computer science. Mississippi has adapted these standards from the nationally developed *Computer Science Teachers Association K-12 Computer Science Standards, Revised 2017*.
mdek12.org

International Society for Technology in Education Standards (ISTE)
Reprinted with permission from *ISTE Standards for Students* (2016). All rights reserved. Permission does not constitute an endorsement by ISTE (iste.org).

College- and Career-Readiness Standards
College- and career-readiness standards emphasize critical thinking, teamwork, and problem-solving skills. Students will learn the skills and abilities demanded by the workforce of today and the future. Mississippi adopted Mississippi College- and Career-Readiness Standards (MCCRS) to provide a consistent, clear understanding of what students are expected to learn and so teachers and parents know what they need to do to help them.
mdek12.org/oae/college-and-career-readiness-standards

Framework for 21st Century Learning
In defining 21st-century learning, the Partnership for 21st Century Skills has embraced key themes and skill areas that represent the essential knowledge for the 21st century: global awareness; financial, economic, business, and entrepreneurial literacy; civic literacy; health literacy; environmental literacy; learning and innovation skills; information, media, and technology skills; and life and career skills.
battelleforkids.org/networks/p21/frameworks-resources

Preface

Secondary CTE programs in Mississippi face many challenges resulting from sweeping educational reforms at the national and state levels. Schools and teachers are increasingly being held accountable for providing applied learning activities to every student in the classroom. This accountability is measured through increased requirements for mastery and attainment of competency as documented through both formative and summative assessments. This document provides information, tools, and solutions that will aid students, teachers, and schools in creating and implementing applied, interactive, and innovative lessons. Through best practices, alignment with national standards and certifications, community partnerships, and a hands-on, student-centered concept, educators will be able to truly engage students in meaningful and collaborative learning opportunities.

The courses in this document reflect the statutory requirements as found in Section 37-3-49, *Mississippi Code of 1972*, as amended (Section 37-3-46). In addition, this curriculum reflects guidelines imposed by federal and state mandates (Laws, 1988, Ch. 487, §14; Laws, 1991, Ch. 423, §1; Laws, 1992, Ch. 519, §4 eff. from and after July 1, 1992; Strengthening Career and Technical Education for the 21st Century Act, 2019 [Perkins V]; and Every Student Succeeds Act, 2015).

Mississippi Teacher Professional Resources

The following are resources for Mississippi teachers:

Curriculum, Assessment, Professional Learning
: Program resources can be found at the RCU's website, rcu.msstate.edu.

Learning Management System: An Online Resource
: Learning management system information can be found on the RCU's website, under Professional Learning.

Should you need additional instructions, contact the RCU at 662.325.2510 or helpdesk@rcu.msstate.edu.

Executive Summary

Pathway Description
The Cyber Foundations I program is designed to provide students with essential skills in computer science, digital literacy, and cybersecurity, laying the groundwork for more advanced studies in the IT field. Students will explore topics such as problem-solving, programming, online safety, and the basics of computer hardware and software.

College, Career, and Certifications
This course aligns with college and career readiness standards, preparing students for further education in computer science and related fields. It provides a foundational understanding that is essential for certifications in areas such as IT Fundamentals, CompTIA, and others related to cybersecurity and digital literacy.

Grade Level and Class Size Recommendations
It is recommended that students enter this program as 6th, 7th, or 8th graders. Exceptions to this are district-level decisions based on class size, enrollment numbers, student maturity, and CTE delivery method. This is a classroom-based course. Therefore, a maximum of 25 students is recommended for each class, and only one class with the teacher at a time.
For students to experience success in the program, the following student prerequisites are suggested:

Student Prerequisites
For students to experience success in the program, the following student prerequisites are suggested:

1. C or higher in English (the previous year)
2. C or higher in high school-level math (last course taken or the instructor can specify the level of math instruction needed)
3. Instructor approval and Test of Adult Basic Education (TABE) reading score (eighth grade or higher)

or

1. TABE reading and math score (eighth grade or higher)
2. Instructor approval

or

1. Instructor approval

Assessment
The latest assessment blueprint for the curriculum can be found at rcu.msstate.edu/curriculum.

Applied Academic Credit
The latest academic credit information can be found at mdek12.org/ese/approved-course-for-the-secondary-schools.

Teacher Licensure
The latest teacher licensure information can be found at mdek12.org/oel/apply-for-an-educator-license.

Professional Learning
If you have specific questions about the content of any training sessions provided, please contact the RCU at 662.325.2510 or helpdesk@rcu.msstate.edu.

Course Outline

This curriculum consists of one 1-credit course.

Cyber Foundations I—Course Code: 000284

Unit	Unit Title	Hours
1	Orientation, Digital Citizenship, and Keyboarding	10
2	Student Organizations, Leadership, and Career Exploration	2
3	21st Century Toolbox	3
4	Productivity Tools	15
5	Block-Based Programming	30
6	Cybersecurity	15
7	Data, Computers, and Society	20
8	App Design	25
9	Multimedia	20
Total		**140**

Career Pathway Outlook

Overview
The Cyber Foundations course is designed to equip students with fundamental skills in computer science, digital literacy, and information technology. This course provides a broad introduction to various aspects of computing, including problem-solving, programming, and understanding the impact of technology on society. Students are prepared for further study in computer science and related fields, as well as for potential careers that leverage these essential skills.

Needs of the Future Workforce
The following data highlights key projected job opportunities in Mississippi from the U.S. Census Bureau, the U.S. Bureau of Labor Statistics (BLS), and the Mississippi Department of Employment Security (MDES):

Table 1.1: Current and Projected Occupation Report

Description	Jobs, 2020	Projected Jobs, 2030	Change (Number)	Change (Percent)	Average Hourly Earnings, 2024
Computer Programmers	470	460	(10)	(2.1%)	$33.53
Web Developers	200	210	10	5%	$31.34
IT Support Specialists	1,040	1,060	20	1.9%	$27.83
Data Analysts	1,820	1,870	50	2.7%	$42.19
Network Administrators	1,630	1,650	20	1.2%	$38.96

Source: Mississippi Department of Employment Security; mdes.ms.gov (2024).

Perkins V Requirements and Academic Infusion
The Cyber Foundations curriculum meets Perkins V requirements by introducing students to foundational concepts in computer science and digital literacy. The curriculum includes classroom instruction and hands-on labs, offering students practical experience that prepares them for further study or entry-level positions in various technology-related fields. Additionally, this curriculum is integrated with academic college- and career-readiness standards. Lastly, it focuses on ongoing and meaningful professional development for teachers as well as relationships with industry.

Transition to Postsecondary Education
The latest articulation information for secondary to postsecondary can be found at the Mississippi Community College Board website, mccb.edu.

Best Practices

Innovative Instructional Technologies

Classrooms should be equipped with tools that will teach today's digital learners through applicable and modern practices. The Cyber Foundations I educator's goal should be to include teaching strategies that incorporate current technology. To make use of the latest online communication tools—wikis, blogs, podcasts, and social media platforms, for example—the classroom teacher is encouraged to use a learning management system that introduces students to education in an online environment and places more of the responsibility of learning on the student.

Differentiated Instruction

Students learn in a variety of ways, and numerous factors—students' background, emotional health, and circumstances, for example—create unique learners. By providing various teaching and assessment strategies, students with various learning preferences can have more opportunities to succeed.

CTE Student Organizations

Teachers should investigate opportunities to sponsor a student organization. There are several here in Mississippi that will foster the types of learning expected from the Cyber Foundations I curriculum. TSA and FBLA are examples of student organizations with many outlets for computer science. Student organizations provide participants and members with growth opportunities and competitive events. They also open the doors to the world of computer science careers and scholarship opportunities.

Cooperative Learning

Cooperative learning can help students understand topics when independent learning cannot. Therefore, you will see several opportunities for group work in the Cyber Foundations I curriculum. To function in today's workforce, students need to be able to work collaboratively with others and solve problems without excessive conflict. The Cyber Foundations I curriculum provides opportunities for students to work together and help each other complete complex tasks. There are many field experiences within the Cyber Foundations I curriculum that will allow and encourage collaboration with professionals currently in the Cyber Foundations field.

Professional Organizations

For students:
Future Business Leaders of America (FBLA)
fbla.org

Technology Student Association (TSA)
tsaweb.org

For teachers:
Association for Career and Technical Education (ACTE)
acteonline.org

Mississippi Educational Computing Association (MECA)
www.ms-meca.org

Using This Document

Competencies and Suggested Objectives
A competency represents a general concept or performance that students are expected to master as a requirement for satisfactorily completing a unit. Students will be expected to receive instruction on all competencies. The suggested objectives represent the enabling and supporting knowledge and performances that will indicate mastery of the competency at the course level.

Teacher Resources
All teachers should request to be added to the Canvas Resource Guide for their course. For questions or to be added to the guide, send a Help Desk ticket to the RCU by emailing helpdesk@rcu.msstate.edu.

Perkins V Quality Indicators and Enrichment Material
Some of the units may include an enrichment section at the end. This material will greatly enhance the learning experiences of students. If the Cyber Foundations I program is using a national certification, work-based learning, or another measure of accountability that aligns with Perkins V as a quality indicator, this material could very well be assessed on that quality indicator. It is the responsibility of the teacher to ensure all competencies for the selected quality indicator are covered throughout the year.

Unit 1: Orientation, Digital Citizenship, and Keyboarding

Competencies and Suggested Objectives
1. Understand school policies, program policies, and safety procedures related to Cyber Foundations. DOK 1 a. Review the school handbook, the technology acceptable use policy, and other safety procedures for building-level situations. b. Examine the course outline and discuss its relevance in today's workforce. c. Demonstrate appropriate safety measures related to technology in the computer lab and online environments. d. Ensure all students master a safety test with 100%.
2. Explore social and ethical issues related to digital citizenship, social media, and artificial intelligence (AI). DOK 2 a. Analyze personal media habits and evaluate how much time is spent with different forms of media. b. Debate the pros and cons of social media when used personally, educationally, and professionally. c. Assess user responsibilities to respect others' creative work. d. Develop strategies to determine inappropriate contact and foster positive interactions when collaborating online. e. Reflect on the outcomes of creating different online personalities. f. Investigate cyberbullying behaviors and their impact on individuals and communities. g. Discuss the ethical use of AI in digital interactions and its societal impact.
3. Collaborate effectively with teachers, peers, and course material using a learning management system (LMS). DOK 2 a. Explore online learning environments and understand how they operate among teachers and students. b. Practice proper e-mail etiquette through real-world scenarios. c. Engage in online learning methods such as discussion boards, student journals, blogs, or wikis.
4. Develop and maintain keyboarding skills. DOK 1 a. Develop touch typing techniques to increase accuracy and speed. b. Demonstrate proper posture and hand placement for effective typing. c. Practice typing regularly to improve proficiency. d. Utilize keyboard shortcuts to enhance productivity.
5. Investigate and compare career opportunities within various Mississippi career clusters. DOK 2 a. Research career opportunities for employment in multiple career clusters (e.g., Law, Public Safety, Corrections, and Security; IT; Health Sciences; Business Management). b. Examine the requirements, skills, wages, education, and employment opportunities in at least one career pathway from each explored cluster. c. Connect computer science and ethical knowledge with employment opportunities in various career clusters.

Note: Safety is to be taught as an ongoing part of the program. Students are required to complete a written safety test with 100% accuracy before entering the shop for lab simulations and projects. This test should be documented in each student's file.
Note: This unit will be ongoing throughout the year. Time allotted for this unit will be distributed over the entire year.

Unit 2: Student Organizations, Leadership, and Career Exploration

Competencies and Suggested Objectives
1. Identify opportunities to participate in student organizations related to technology and business. DOK 1 a. Research and list student organizations available at the school for technology and business students (e.g., Future Business Leaders of America [FBLA] and Technology Students Association [TSA]). b. Identify student competitions and events associated with each organization.
2. Understand the structure and procedures of business meetings. DOK 2 a. Demonstrate the process of opening and closing a business meeting. b. Understand and apply the norms and etiquette of conducting business meetings.
3. Develop and demonstrate leadership skills. DOK 2 a. Identify and discuss the characteristics of an effective leader. b. Participate in leadership activities and simulations to practice and enhance leadership skills. c. Evaluate different leadership styles and their effectiveness in various situations.
4. Explore career pathways through involvement in student organizations. DOK 3 a. Research career pathways and opportunities provided by involvement in student organizations. b. Discuss how participation in student organizations can enhance career readiness and networking opportunities.

Note: This unit will be ongoing throughout the year. Time allotted for this unit will be distributed over the entire year.

Unit 3: 21st Century Toolbox

Competencies and Suggested Objectives
1. Differentiate between various learning styles and personality traits found within the classroom and workplace. DOK 2 a. Complete a learning styles inventory to identify individual learning preferences. b. Identify personality traits through a personality self-test. c. Discuss strategies for effective collaboration in diverse work environments, considering different learning styles and personality traits.
2. Demonstrate effective time management skills, study skills, and note-taking strategies. DOK 2 a. Develop short-term and long-term goals, focusing on academic and personal growth. b. Practice and apply effective time management techniques to daily activities. c. Implement and refine study skills and note-taking strategies using technological tools.
3. Demonstrate effective public speaking skills. DOK 2 a. Practice clear and effective communication within group settings. b. Develop and deliver presentations with confidence, ensuring clarity and audience engagement.
4. Demonstrate knowledge of 21st-century skills, including ethical AI and data use. DOK 3 a. Collaborate effectively with peers on projects and assignments, showcasing teamwork and communication skills. b. Demonstrate creativity and imagination in problem-solving scenarios. c. Utilize critical thinking to analyze and solve complex problems. d. Apply problem-solving techniques to various scenarios, emphasizing ethical considerations in AI and data use.
5. Explore career pathways through career exploration activities. DOK 2 a. Research and identify career pathways within various Mississippi career clusters, understanding the required 21st-century skills. b. Discuss the application of these skills to different career paths, emphasizing how they enhance career readiness.
6. Update and refine an Individual Success Plan (ISP) to align with career interests and educational goals. DOK 2 a. Identify the basic components of the ISP, linking it to the 14 national career clusters and secondary and postsecondary education. b. Select and print courses that meet graduation requirements and reflect the ISP, ensuring alignment with career goals.

Note: This unit will be ongoing throughout the year. Time allotted for this unit will be distributed over the entire year.

Unit 4: Productivity Tools

Competencies and Suggested Objectives
1. Develop proficiency in word processing applications. DOK2 a. Create, format, and edit documents using word processing software (e.g., Microsoft Word, Google Docs, Apple Pages). b. Utilize features such as tables, bullet points, headers, and footers to enhance document presentation. c. Save and share documents in various formats (e.g., PDF).
2. Develop proficiency in spreadsheet applications. DOK2 a. Create, format, and edit spreadsheets using spreadsheet software (e.g., Microsoft Excel, Google Sheets, Apple Numbers). b. Use basic formulas (performing mathematical calculations on multiple cells) and functions (Sum, Average, Count) to perform calculations. c. Create charts and graphs to represent data visually. d. Utilize organizational features such as sorting and multiple worksheets.
3. Develop proficiency in presentation software. DOK2 a. Create, format, and edit presentations using presentation software (e.g., Microsoft PowerPoint, Google Slides, Apple Keynote). b. Incorporate text, images, charts, and animations to enhance presentations. c. Practice good design habits (i.e., 5x5 rule). d. Deliver presentations effectively using presentation software.
4. Understand and apply AI basics in productivity software. DOK3 a. Explore using AI tools in word processing, spreadsheet, and presentation software. b. Utilize AI-powered features such as grammar and style suggestions, data analysis, and design recommendations.
5. Develop skills in using chatbots and prompt engineering. DOK3 a. Understand the basics of chatbots and their applications in productivity tools. b. Create and test simple chatbots using available platforms. c. Practice prompt engineering to improve the effectiveness of chatbots.
6. Discuss the ethical use of AI in productivity tools. DOK3 a. Identify ethical considerations when using AI in productivity software. b. Explore real-world examples of ethical and unethical AI use. c. Reflect on how ethical AI use can impact careers and productivity.
7. Explore career pathways related to productivity tools and AI. DOK2 a. Research careers that involve extensive use of productivity tools and AI (e.g., data analyst, digital marketer, administrative professional). b. Discuss how proficiency in these tools can enhance career readiness and opportunities. c. Connect classroom learning to real-world applications and career pathways.

Note: This unit will be ongoing throughout the year. Time allotted for this unit will be distributed over the entire year.

Unit 5: Block-Based Programming

Competencies and Suggested Objectives
1. Understand how programming is used to solve problems. DOK2 a. Explain how computer science and coding are used in various fields to solve problems. b. Break down problems into smaller, manageable steps using algorithms. c. Practice using pseudocode to draft coding solutions.
2. Develop proficiency in block-based programming tools. DOK2 a. Navigate the user interface of multiple block-based programming environments (e.g., Scratch, Code.org, Tynker). b. Create and manipulate sprites and objects in a block-based environment. c. Use block-based commands to control the movement, appearance, and interactions of sprites. d. Create a simple program to manipulate sprites and objects in a block-based environment.
3. Create simple projects using block-based programming. DOK2 a. Design and develop simple animations and interactive applications using block-based programming. b. Implement user interactions and control structures in projects. c. Investigate variables. • Understand and apply the concept of variables within block-based programming. d. Apply Draw Loop. • Implement loops to create repetitive patterns and animations. e. Counter-pattern, velocity, rotation speed. • Utilize variables and control structures to manipulate movement and speed. f. Apply Booleans & Conditionals. • Apply Boolean logic and conditional statements to control program flow. g. Debug and test block-based programs to ensure they work correctly. h. Write pseudocode. • Write pseudocode to plan and structure programs before implementation. i. Debug/Troubleshoot. • Identify and fix errors in block-based programs using debugging techniques.
4. Understand the ethical considerations in programming and AI. DOK2 a. Discuss the importance of ethical behavior in programming and AI development. b. Explore examples of ethical and unethical programming practices. c. Reflect on the impact of ethical programming on society and careers. d. Evaluate code generated by an AI.

5. Explore career pathways related to programming. DOK3
   a. Research programming, computer science, and AI careers within Mississippi's career clusters.
   b. Identify the skills, education, and experience required for various programming careers.
   c. Understand how skills in block-based programming can be applied to real-world jobs.
   d. Connect classroom learning to real-world applications and career pathways.

Note: This unit will be ongoing throughout the year. Time allotted for this unit will be distributed over the entire year.

Unit 6: Cybersecurity

Competencies and Suggested Objectives
1. Understand the importance of cybersecurity. DOK2 a. Define cybersecurity and explain its significance in the digital age. b. Discuss the impact of cyber threats on individuals, organizations, and society.
2. Identify types of cyber threats and attacks. DOK2 a. Recognize various types of cyber threats, including malware, phishing, and ransomware. b. Explain how cyber-attacks are carried out and their potential consequences.
3. Learn basic cybersecurity practices and protection methods. DOK2 a. Demonstrate how to create strong passwords and manage them securely. b. Understand the importance of software updates and patches. c. Identify safe browsing practices and how to avoid phishing scams.
4. Understand the CIA Triad and its relevance to cybersecurity. DOK2 a. Explain the components of the CIA Triad: Confidentiality, Integrity, and Availability. b. Discuss how the CIA Triad is used to protect information systems. c. Apply the CIA Triad principles to real-world cybersecurity scenarios.
5. Develop adversarial thinking skills. DOK2 a. Explain the concept of adversarial thinking in cybersecurity. b. Analyze potential threats and vulnerabilities in information systems. c. Develop strategies to mitigate and defend against cyber-attacks.
6. Understand encryption and encoding in digital security. DOK3 a. Explore the origins of common ciphers (i.e., Pigpen, Caesar, Substitution). b. Use encoding methods to represent and secure information (i.e., Pigpen, Caesar, Substitution). c. Compare and contrast asymmetric and symmetric encryption.
7. Explore AI in cybersecurity. DOK3 a. Understand the role of AI in identifying and mitigating cyber threats. b. Explore AI tools and techniques used in cybersecurity, such as anomaly detection and threat intelligence. c. Discuss the benefits and challenges of using AI in cybersecurity.
8. Understand ethical considerations in cybersecurity. DOK3 a. Discuss the ethical implications of cybersecurity practices and policies. b. Explore real-world examples of ethical and unethical behavior in cybersecurity. c. Reflect on the role of ethics in protecting information and privacy.
9. Explore career pathways related to cybersecurity across various career clusters. DOK3 a. Research careers in cybersecurity within various Mississippi career clusters (e.g., Business Management, Health Sciences, Law and Public Safety). b. Identify the skills, education, and experience required for various cybersecurity roles. c. Understand how cybersecurity skills can be applied to real-world jobs in different fields. d. Connect classroom learning to real-world applications and career pathways.

Unit 7: Data, Computers, and Society

Competencies and Suggested Objectives
1. Examine data collection and representation using the problem-solving process. DOK2 a. Understand data as information collected from the world to help make a recommendation or solve a problem. b. Provide examples of how different representations of data can affect its ability to solve problems. c. Choose the most effective way to represent information based on its intended use. d. Describe essential features of a system for representing information. e. Create, use, and provide feedback on a system for representing information. f. Iteratively improve a system for representing information by testing and responding to feedback.
2. Identify and design ASCII and binary systems. DOK4 a. Define terms associated with ASCII and binary systems. b. Use the ASCII system to encode and decode text information in binary. c. Describe common features of systems used to represent information in binary. d. Use a binary system to represent numbers. Extend a representation system based on patterns.
3. Apply concepts to solve problems using data. DOK3 a. Use the problem-solving process to answer questions using data. b. Identify and collect relevant data to help solve a problem. c. Use data to draw conclusions.
4. Investigate how data is collected. DOK3 a. Give examples of how data is collected from sensors and by tracking user behavior. b. Determine which data would be helpful in solving a problem and how to collect it. c. Distinguish between data that users intentionally and unintentionally produce.
5. Analyze and revise data to make it useful. DOK2 a. Identify and remove irrelevant data from a data set. b. Create a bar chart based on a set of data. c. Explain why a set of data must be cleaned before a computer can use it.
6. Critique data to make and support decisions. DOK3 a. Use tables and visualizations summarizing data to support a decision. b. Present and critique interpretations of tables and visualizations. c. Identify additional data that could improve a decision. d. Organize data to support a claim. e. Find patterns and relationships in data.
7. Construct a plan to automate data decisions. DOK3 a. Design an algorithm for making decisions using data as inputs. b. Explain the benefits and drawbacks of using computers for automated decision-making. c. Interpret collected data to identify patterns.
8. Apply concepts of data collection and interpretation to make a recommendation. DOK3 a. Apply the data problem-solving process to a personally relevant topic. b. Determine appropriate sources of data needed to solve a problem.

9. Explore AI in data science. DOK3 a. Understand the role of AI in data collection, analysis, and interpretation. b. Explore AI tools and techniques used in data science, such as machine learning and data mining. c. Discuss the ethical implications of using AI in data science.
10. Investigate career opportunities in various career clusters related to data. DOK2 a. Research career opportunities for employment in various career clusters (e.g., cybersecurity, genetics, business). b. Examine the requirements, skills, wages, education, and employment opportunities in at least one career pathway from different career clusters. c. Discuss how computer science impacts various career clusters.

Unit 8: App Design

Competencies and Suggested Objectives
1. Compare and contrast different types of applications. DOK2 a. Identify ways in which apps can affect social change. b. Identify the user needs addressed by different types of apps.
2. Identify and examine user needs to understand the purposes of design. DOK2 a. Express opinions respectfully and effectively. b. Critically evaluate an object for how well its design meets a given set of needs. c. Recognize empathy for the user as an important component of the design process. d. Distinguish between creator needs and user needs.
3. Develop paper prototypes to test ideas and assumptions. DOK3 a. Use a paper prototype to test an app before programming it. b. Identify the user needs a prototype was designed to address. c. Categorize and prioritize user feedback for an app. d. Create a paper prototype for the screens of an app. e. Design the functionality of an app to address specific user needs. f. Identify improvements to an app based on user testing. g. Design the user interface of an app.
4. Develop a digital prototype of an app. DOK3 a. Construct transformations of graphic designs. b. Construct graphic animations. c. Generate graphics and animations for the app.
5. Revise and formulate improvements based on user feedback. DOK3 a. Develop a detailed plan for testing the prototype. b. Collect and analyze test data. c. Revise and improve the app based on testing results. d. Debug/Troubleshoot: Identify and fix errors in the app prototype.
6. Integrate AI elements into app design. DOK3 a. Explore AI tools and techniques that can be integrated into app development. b. Understand the role of AI in enhancing user experience and app functionality. c. Discuss the ethical implications of using AI in apps.
7. Investigate career opportunities in the software development and engineering career clusters. DOK2 a. Research career opportunities in the software development or engineering career clusters. b. Examine the requirements, skills, wages, education, and employment opportunities in at least one career pathway in these clusters. c. Discuss how computer science impacts the software development and engineering career clusters.

Unit 9: Multimedia

Competencies and Suggested Objectives
1. Understand the principles of graphic design. DOK2 a. Explain the basic principles of graphic design, including balance, contrast, alignment, repetition, and proximity. b. Identify effective use of color, typography, and imagery in graphic design. c. Evaluate the visual impact of different design elements.
2. Create digital graphics and multimedia content. DOK2 a. Use graphic design software to create digital graphics and visual content. b. Develop multimedia content, including images, videos, and animations. c. Integrate multimedia elements into a cohesive digital project.
3. Explore modern web development using drag-and-drop website builders. DOK2 a. Understand the basics of web design and layout. b. Use drag-and-drop website builders to create functional and visually appealing websites. c. Customize templates and design elements to meet specific needs.
4. Integrate multimedia and graphic design into web development. DOK3 a. Incorporate digital graphics and multimedia content into web pages. b. Ensure that multimedia elements enhance the user experience. c. Optimize multimedia content for web performance.
5. Utilize AI in digital design and multimedia. DOK3 a. Explore AI tools and techniques used in graphic design and multimedia creation. b. Use AI to enhance design elements, such as automated layout suggestions and image editing. c. Discuss the ethical implications of using AI in digital design and multimedia.
6. Understand the ethical and legal considerations in digital design. DOK3 a. Discuss the ethical implications of digital content creation and distribution. b. Explore copyright laws and fair use guidelines related to digital design. c. Reflect on the importance of ethical behavior in digital design and multimedia creation.
7. Investigate career opportunities in multimedia, graphic design, and web development. DOK2 a. Research career opportunities in multimedia, graphic design, and web development. b. Examine the requirements, skills, wages, education, and employment opportunities in at least one career pathway from these fields. c. Discuss how digital design skills impact various career clusters.

Enhancement Unit

Competencies and Suggested Objectives
Unit 1 Orientation (Digital Literacy, Career Exploration, Student Organizations, Keyboarding) Competency: Advanced Digital Literacy DOK3 • Objective: Evaluate the credibility of various digital information sources and distinguish between reliable and unreliable information. Competency: In-Depth Career Exploration DOK3 • Objective: Conduct comprehensive research on career pathways in multiple industries, including detailed analysis of job trends and future outlooks.
Unit 2 Student Organizations and Leadership Competency: Advanced Leadership Skills DOK3 • Objective: Design and implement a leadership project within a student organization, demonstrating effective leadership and project management skills. Competency: Strategic Planning for Student Organizations DOK3 • Objective: Develop a strategic plan for a student organization, including goal setting, resource allocation, and long-term planning.
Unit 3 21st Century Toolbox Competency: Advanced Data Management DOK3 • Objective: Implement data management techniques using advanced spreadsheet functions, including pivot tables and complex formulas. Competency: AI-Enhanced Productivity DOK3 • Objective: Integrate AI tools into productivity software to automate repetitive tasks and enhance efficiency.
Unit 4 Productivity Tools Competency: Mastery of Advanced Productivity Software Features DOK3 • Objective: Utilize advanced features of productivity software (e.g., macros in Excel, advanced styles in Word) to create sophisticated documents and presentations. Competency: AI in Productivity DOK3 • Objective: Implement AI tools in productivity applications to optimize workflow and improve document management.
Unit 5 Block-Based Programming Competency: Transition to Text-Based Programming DOK3 • Objective: Develop proficiency in a text-based programming language by translating block-based programs into text-based code. Competency: Advanced Problem-Solving with Programming DOK3 • Objective: Design and implement complex algorithms in a block-based programming environment, focusing on optimization and efficiency.

Unit 6 Cybersecurity

Competency: Advanced Cyber Defense Techniques DOK3

- Objective: Implement advanced cyber defense techniques, including intrusion detection systems and incident response strategies.

Competency: AI in Cybersecurity DOK3

- Objective: Utilize AI tools for real-time threat detection and mitigation, exploring machine learning algorithms for cybersecurity applications.

Unit 7 Data, Computers, and Society

Competency: Advanced Data Analysis DOK3

- Objective: Perform advanced data analysis using statistical software, focusing on predictive analytics and data visualization techniques.

Competency: AI in Data Science DOK3

- Objective: Develop machine learning models to analyze large datasets, applying AI techniques to derive insights and make data-driven decisions.

Unit 8 App Design

Competency: Advanced User Experience (UX) Design DOK3

- Objective: Conduct comprehensive user research and usability testing to refine app designs, focusing on creating intuitive and accessible user interfaces.

Competency: AI Integration in App Development DOK3

- Objective: Integrate AI features into app development, such as natural language processing for chatbots and recommendation systems.

Unit 9 Multimedia

Competency: Advanced Web Development DOK3

- Objective: Develop websites using programming languages such as HTML, CSS, and JavaScript, incorporating advanced features and responsive design techniques.

Competency: Multimedia Content Creation DOK3

- Objective: Create complex multimedia projects using advanced tools and techniques, integrating video, audio, animation, and interactive elements.

Student Competency Profile

Student's Name: _____

This record is intended to serve as a method of noting student achievement of the competencies in each unit. It can be duplicated for each student, and it can serve as a cumulative record of competencies achieved in the course.

In the blank before each competency, place the date on which the student mastered the competency.

Unit 1: Orientation, Digital Citizenship, and Keyboarding		
	1.	Understand school policies, program policies, and safety procedures related to Cyber Foundations.
	2.	Explore social and ethical issues related to digital citizenship, social media, and artificial intelligence (AI).
	3.	Collaborate effectively with teachers, peers, and course material using a learning management system.
	4.	Develop and maintain keyboarding skills.
	5.	Investigate and compare career opportunities within various Mississippi career clusters.
Unit 2: Student Organizations, Leadership, and Career Exploration		
	1.	Identify opportunities to participate in student organizations related to technology and business.
	2.	Understand the structure and procedures of business meetings.
	3.	Develop and demonstrate leadership skills.
	4.	Explore career pathways through involvement in student organizations.
Unit 3: 21st Century Toolbox		
	1.	Differentiate between various learning styles and personality traits found within the classroom and workplace.
	2.	Demonstrate effective time management skills, study skills, and note-taking strategies.
	3.	Demonstrate effective public speaking skills.
	4.	Demonstrate knowledge of 21st-century skills, including ethical AI and data use.
	5.	Explore career pathways through career exploration activities.
	6.	Update and refine an Individual Success Plan (ISP) to align with career interests and educational goals.
Unit 4: Productivity Tools		
	1.	Develop proficiency in word processing applications.
	2.	Develop proficiency in spreadsheet applications.

	3.	Develop proficiency in presentation software.
	4.	Understand and apply AI basics in productivity software.
	5.	Develop skills in using chatbots and prompt engineering.
	6.	Discuss the ethical use of AI in productivity tools.
	7.	Explore career pathways related to productivity tools and AI.
Unit 5: Block-Based Programming		
	1.	Understand how programming is used to solve problems.
	2.	Develop proficiency in block-based programming tools.
	3.	Create simple projects using block-based programming.
	4.	Understand the ethical considerations in programming and AI.
	5.	Explore career pathways related to programming.
Unit 6: Cybersecurity		
	1.	Understand the importance of cybersecurity.
	2.	Identify types of cyber threats and attacks.
	3.	Learn basic cybersecurity practices and protection methods.
	4.	Understand the CIA Triad and its relevance to cybersecurity.
	5.	Develop adversarial thinking skills.
	6.	Understand encryption and encoding in digital security.
	7.	Explore AI in cybersecurity.
	8.	Understand ethical considerations in cybersecurity.
	9.	Explore career pathways related to cybersecurity across various career clusters.
Unit 7: Data, Computers, and Society		
	1.	Examine data collection and representation using the problem-solving process.
	2.	Identify and design ASCII and binary systems.
	3.	Apply concepts to solve problems using data.
	4.	Investigate how data is collected.
	5.	Analyze and revise data to make it useful.
	6.	Critique data to make and support decisions.
	7.	Construct a plan to automate data decisions.
	8.	Apply concepts of data collection and interpretation to make a recommendation.
	9.	Explore AI in data science.
	10.	Investigate career opportunities in various career clusters related to data.
Unit 8: App Design		
	1.	Compare and contrast different types of apps.

	2.	Identify and examine user needs to understand the purposes of design.
	3.	Develop paper prototypes to test ideas and assumptions.
	4.	Develop a digital prototype of an app.
	5.	Revise and formulate improvements based on user feedback.
	6.	Integrate AI elements into app design.
	7.	Investigate career opportunities in the software development and engineering career clusters.
Unit 9: Multimedia		
	1.	Understand the principles of graphic design.
	2.	Create digital graphics and multimedia content.
	3.	Explore modern web development using drag-and-drop website builders.
	4.	Integrate multimedia and graphic design into web development.
	5.	Utilize AI in digital design and multimedia.
	6.	Understand the ethical and legal considerations in digital design.
	7.	Investigate career opportunities in multimedia, graphic design, and web development.

Appendix A: Industry Standards

	Units	1	2	3	4	5	6	7	8	9
Standards										
CS1		X	X							
CS2		X	X							
CS3			X		X					
CS4			X							
CS5										
CS6			X	X		X				
CS7			X	X	X	X	X		X	
CS8			X		X	X	X		X	
CS9			X	X	X	X	X	X	X	X
CS10										
CS11					X	X	X		X	
CS12		X								
CS13				X						
CS14				X						
CS15							X			X
CS16		X			X	X	X		X	

CSS1-21st Century Themes

CS1 Global Awareness

1. Using 21st-century skills to understand and address global issues
2. Learning from and working collaboratively with individuals representing diverse cultures, religions, and lifestyles in a spirit of mutual respect and open dialogue in personal, work, and community contexts
3. Understanding other nations and cultures, including the use of non-English languages

CS2 Financial, Economic, Business, and Entrepreneurial Literacy

1. Knowing how to make appropriate personal economic choices
2. Understanding the role of the economy in society
3. Using entrepreneurial skills to enhance workplace productivity and career options

CS3 Civic Literacy

1. Participating effectively in civic life through knowing how to stay informed and understanding governmental processes
2. Exercising the rights and obligations of citizenship at local, state, national, and global levels
3. Understanding the local and global implications of civic decisions

CS4 Health Literacy

1. Obtaining, interpreting, and understanding basic health information and services and using such information and services in ways that enhance health
2. Understanding preventive physical and mental health measures, including proper diet, nutrition, exercise, risk avoidance, and stress reduction
3. Using available information to make appropriate health-related decisions
4. Establishing and monitoring personal and family health goals
5. Understanding national and international public health and safety issues

CS5 Environmental Literacy

1. Demonstrating knowledge and understanding of the environment and the circumstances and conditions affecting it, particularly as relates to air, climate, land, food, energy, water, and ecosystems
2. Demonstrating knowledge and understanding of society's impact on the natural world (e.g., population growth, population development, resource consumption rate, etc.)
3. Investigating and analyzing environmental issues and making accurate conclusions about effective solutions
4. Taking individual and collective action toward addressing environmental challenges (e.g., participating in global actions, designing solutions that inspire action on environmental issues)

CSS2-Learning and Innovation Skills

CS6 Creativity and Innovation
1. Think creatively
2. Work creatively with others
3. Implement innovations

CS7 Critical Thinking and Problem Solving
1. Reason effectively
2. Use systems thinking
3. Make judgments and decisions
4. Solve problems

CS8 Communication and Collaboration
1. Communicate clearly
2. Collaborate with others

CSS3-Information, Media, and Technology Skills

CS9 Information Literacy
1. Access and evaluate information
2. Use and manage information

CS10 Media Literacy
1. Analyze media
2. Create media products

CS11 ICT Literacy
1. Apply technology effectively

CSS4-Life and Career Skills

CS12 Flexibility and Adaptability
1. Adapt to change
2. Be flexible

CS13 Initiative and Self-Direction
1. Manage goals and time
2. Work independently
3. Be self-directed learners

CS14 Social and Cross-Cultural Skills
1. Interact effectively with others

2. Work effectively in diverse teams

CS15 Productivity and Accountability

1. Manage projects
2. Produce results

CS16 Leadership and Responsibility

- Guide and lead others
- Be responsible to others

Appendix B: ISTE Standards

	Unit	1	2	3	4	5	6	7	8	9
Standards										
T1		X		X	X	X	X	X	X	
T2		X								
T3		X	X	X	X			X		X
T4			X	X	X	X				
T5						X		X	X	X
T6				X	X					
T7			X							

International Society for Technology in Education (ISTE)

T1 Empowered Learner

Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.

a. Articulate and set personal learning goals, develop strategies leveraging technology to achieve them, and reflect on the learning process itself to improve learning outcomes.
b. Build networks and customize their learning environments in ways that support the learning process.
c. Use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.
d. Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use, and troubleshoot current technologies, and are able to transfer their knowledge to explore emerging technologies.

T2 Digital Citizen

Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal, and ethical.

a. Cultivate and manage their digital identity and reputation and are aware of the permanence of their actions in the digital world.
b. Engage in positive, safe, legal, and ethical behavior when using technology, including social interactions online or when using networked devices.
c. Demonstrate an understanding of and respect for the rights and obligations of using and sharing intellectual property.
d. Manage their personal data to maintain digital privacy and security and are aware of data-collection technology used to track their navigation online.

T3 Knowledge Constructor

Students critically curate a variety of resources using digital tools to construct knowledge, produce creative artifacts, and make meaningful learning experiences for themselves and others.

a. Plan and employ effective research strategies to locate information and other resources for their intellectual or creative pursuits.
b. Evaluate the accuracy, perspective, credibility, and relevance of information, media, data or other resources.

c. Curate information from digital resources using a variety of tools and methods to create collections of artifacts that demonstrate meaningful connections or conclusions.
d. Build knowledge by actively exploring real-world issues and problems, developing ideas and theories, and pursuing answers and solutions.

T4 Innovative Designer

Students use a variety of technologies within a design process to identify and solve problems by creating new, useful, or imaginative solutions.

a. Know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.
b. Select and use digital tools to plan and manage a design process that considers design constraints and calculated risks.
c. Develop, test, and refine prototypes as part of a cyclical design process.
d. Exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.

T5 Computational Thinker

Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.

a. Formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models, and algorithmic thinking in exploring and finding solutions.
b. Collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making.
c. Break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
d. Understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.

T6 Creative Communicator

Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats, and digital media appropriate to their goals.

a. Choose the appropriate platforms and tools for meeting the desired objectives of their creation or communication.
b. Create original works or responsibly repurpose or remix digital resources into new creations.
c. Communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models, or simulations.
d. Publish or present content that customizes the message and medium for their intended audiences.

T7 Global Collaborator

Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally.

a. Use digital tools to connect with learners from a variety of backgrounds and cultures, engaging with them in ways that broaden mutual understanding and learning.

b. Use collaborative technologies to work with others, including peers, experts, or community members, to examine issues and problems from multiple viewpoints.
c. Contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.
d. Explore local and global issues and use collaborative technologies to work with others to investigate solutions.

Appendix C: Mississippi College- and Career-Readiness Standards for Computer Science (MS CCRS)

	Unit	1	2	3	4	5	6	7	8	9
Standards										
CS.2.1		X			X					
CS.2.2					X				X	
CS.2.3							X			
NI.2.1						X	X			
NI.2.2						X	X			
NI.2.3							X			
DA.2.1					X		X	X		
DA.2.2				X						
DA.2.3							X			
AP.2.1						X				
AP.2.2						X				
AP.2.3						X				
AP.2.4						X				
AP.2.5						X				
AP.2.6			X							
AP.2.7						X				
AP.2.8						X				
AP.2.9			X							
AP.2.10						X				
IC.2.1		X			X				X	
IC.2.2									X	
IC.2.3			X							
IC.2.4							X			
CS.3A.1					X					
CS.3A.2					X					
CS.3A.3										X
NI.3A.1								X		
NI.3A.2							X			
NI.3A.3							X			
DA.3A.1				X				X		
DA.3A.2				X						
DA.3A.3				X						
DA.3A.4								X		
AP.3A.1						X				
AP.3A.2						X				
AP.3A.3						X				
AP.3A.4						X				
AP.3A.5						X				
AP.3A.6						X				
AP.3A.7			X			X				
AP.3A.8						X				
AP.3A.9						X				
AP.3A.10			X			X				
AP.3A.11						X				
IC.3A.1		X			X				X	
IC.3A.2					X				X	
IC.3A.3								X		
IC.3A.4								X		
IC.3A.5									X	
IC.3A.6								X		
IC.3A.7								X		

Level 2: GRADES 6-8 - Computing Systems (CS)

CS.2 Computing Systems

Conceptual understanding: People interact with a wide variety of computing devices that collect, store, analyze, and act upon information in ways that can affect human capabilities both positively and negatively. The physical components (hardware) and instructions (software) that make up a computing system communicate and process information in digital form. An understanding of hardware and software is useful when troubleshooting a computing system that does not work as intended.

2.1 Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices. [DEVICES] (P3.3)

The study of human-computer interaction (HCI) can improve the design of devices, including both hardware and software.

a. Students should make recommendations for existing devices (e.g., a laptop, phone, or tablet) or design their own components or interface (e.g., create their own controllers). Teachers can guide students to consider usability through several lenses, including accessibility, ergonomics, and learnability. For example, assistive devices provide capabilities such as scanning written information and converting it to speech.

2.2 Design projects that combine hardware and software components to collect and exchange data. [HARDWARE & SOFTWARE] (P5.1)

Collecting and exchanging data involves input, output, storage, and processing. When possible, students should select the hardware and software components for their project designs by considering factors such as functionality, cost, size, speed, accessibility, and aesthetics.

a. Students will design projects that use both hardware and software to collect and exchange data. For example, components for a mobile app could include an accelerometer, GPS, and speech recognition. The choice of a device that connects wirelessly through a Bluetooth connection versus a physical USB connection involves a tradeoff between mobility and the need for an additional power source for the wireless device.

2.3 Systematically identify and fix problems with computing devices and their components. [TROUBLESHOOTING] (P6.2)

Since a computing device may interact with interconnected devices within a system, problems may not be due to the specific computing device itself but to devices connected to it.

a. Students will use a structured process to troubleshoot problems with computing systems and ensure that potential solutions are not overlooked. Examples of troubleshooting strategies include following a troubleshooting flow diagram, making changes to software to see if hardware will work, checking connections and settings, and swapping in working components.

Level 2: GRADES 6-8 - Networks and the Internet

NI.2 Networks and the Internet

Conceptual Understanding: 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 by providing fast, secure communication and facilitating innovation.

2.1 Model the role of protocols in transmitting data across networks and the Internet. [NETWORK COMMUNICATION & ORGANIZATION] (P4.4)

Protocols are rules that define how messages between computers are sent. They determine how quickly and securely information is transmitted across networks and the Internet, as well as how to handle errors in transmission.

a. Students should model how data is sent using protocols to choose the fastest path, to deal with missing information, and to deliver sensitive data securely. For example, students could devise a plan for resending lost information or for interpreting a picture that has missing pieces. The priority at this grade level is understanding the purpose of protocols and how they enable secure and errorless communication. Knowledge of the details of how specific protocols work is not expected.

2.2 Explain how physical and digital security measures protect electronic information. [CYBERSECURITY] (P7.2)

Information that is stored online is vulnerable to unwanted access. Examples of physical security measures to protect data include keeping passwords hidden, locking doors, making backup copies on external storage devices, and erasing a storage device before it is reused. Examples of digital security measures include secure router admin passwords, firewalls that limit access to private networks, and the use of a protocol, such as HTTPS, to ensure secure data transmission.

a. Students will explain how physical and digital security measures protect electronic information.

2.3 Apply multiple methods of encryption to model the secure transmission of information. [CYBERSECURITY] (P4.4)

Encryption can be as simple as letter substitution or as complicated as modern methods used to secure networks and the Internet.

a. Students should encode and decode messages using a variety of encryption methods, and they should understand the different levels of complexity used to hide or secure information. For example, students could secure messages using methods like Caesar cyphers or steganography (i.e., hiding messages inside a picture or other data). They can also model more complicated methods, such as public key encryption, through unplugged activities.

Level 2: GRADES 6-8 - Data and Analysis

DA.2 Data and Analysis

Conceptual Understanding: Computing systems exist to process data. The amount of digital data generated in the world is rapidly expanding, so the need to process data effectively is increasingly important. Data is collected and stored so that it can be analyzed to better understand the world and make more accurate predictions.

2.1 Represent data using multiple encoding schemes. [STORAGE] (P4.0)

Data representations occur at multiple levels of abstraction, from the physical storage of bits to the arrangement of information into organized formats (e.g., tables).

a. Students should represent the same data in multiple ways. For example, students could represent the same color using binary, RGB values, hex codes (low-level

representations), as well as forms understandable by people, including words, symbols, and digital displays of the color (high-level representations).

2.2 Collect data using computational tools and transform the data to make it more useful and reliable. [COLLECTION, VISUALIZATION, & TRANSFORMATION] (P6.3)

As students continue to build on their ability to organize and present data visually to support a claim, they will need to understand when and how to transform data for this purpose.

a. Students should transform data to remove errors, highlight or expose relationships, and/or make it easier for computers to process. The cleaning of data is an important transformation for ensuring consistent format and reducing noise and errors (e.g., removing irrelevant responses in a survey). An example of a transformation that highlights a relationship is representing males and females as percentages of a whole instead of as individual counts.

2.3 Refine computational models based on the data they have generated. [INFERENCE & MODELS] (P5.3, P4.4)

A model may be a programmed simulation of events or a representation of how various data are related.

a. Students will refine computational models by considering which data points are relevant, how data points relate to each other, and if the data is accurate. For example, students may make a prediction about how far a ball will travel based on a table of data related to the height and angle of a track. The students could then test and refine their model by comparing predicted versus actual results and considering whether other factors are relevant (e.g., size and mass of the ball). Additionally, students could refine game mechanics based on test outcomes in order to make the game more balanced or fair.

Level 2: GRADES 6-8 - Algorithms and Programming

AP.2 Algorithms and Programming

Conceptual understanding: An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are translated into programs, or code, to provide instructions for computing devices. Algorithms and programming control all computing systems, empowering people to communicate with the world in new ways and solve compelling problems. The development process to create meaningful and efficient programs involves choosing which information to use and how to process and store it, breaking apart large problems into smaller ones, recombining existing solutions, and analyzing different solutions.

2.1 Use flowcharts and/or pseudocode to address complex problems as algorithms. [ALGORITHMS] (P4.4, P4.1)

Complex problems are problems that would be difficult for students to solve computationally.

a. Students will use pseudocode and/or flowcharts to organize and sequence an algorithm that addresses a complex problem, even though they may not actually program the solutions. For example, students might express an algorithm that produces a recommendation for purchasing sneakers based on inputs such as size, colors, brand, comfort, and cost. Testing the algorithm with a wide range of inputs

and users allows students to refine their recommendation algorithm and to identify other inputs they may have initially excluded.

2.2 Create clearly named variables that represent different data types and perform operations on their values. [VARIABLES] (P5.1, P5.2)

A variable is like a container with a name, in which the contents may change, but the name (identifier) does not.

a. When planning and developing programs, students should decide when and how to declare and name new variables. Examples of operations include adding points to the score, combining user input with words to make a sentence, changing the size of a picture, or adding a name to a list of people.
b. Students should use naming conventions to improve program readability.

2.3 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. [CONTROL] (P5.1, P5.2)

Control structures can be combined in many ways. Nested loops are loops placed within loops. Compound conditionals combine two or more conditions in a logical relationship (e.g., using AND, OR, and NOT), and nesting conditionals within one another allows the result of one conditional to lead to another.

a. Students will design and develop programs that combine control structures. For example, when programming an interactive story, students could use a compound conditional within a loop to unlock a door only if a character has a key AND is touching the door.

2.4 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. [MODULARITY] (P3.2)

Decomposition facilitates aspects of program development by allowing students to focus on one piece at a time (e.g., getting input from the user, processing the data, and displaying the result to the user). Decomposition also enables different students to work on different parts at the same time.

a. Students should break down problems into subproblems, which can be further broken down to smaller parts. For example, animations can be decomposed into multiple scenes, which can be developed independently.

2.5 Create procedures with parameters to organize code and make it easier to reuse. [MODULARITY] (P4.1, P4.3)

a. Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. These procedures can be generalized by defining parameters that create different outputs for a wide range of inputs. For example, a procedure to draw a circle involves many instructions, but all of them can be invoked with one instruction, such as “drawCircle.” By adding a radius parameter, the user can easily draw circles of different sizes.

2.6 Seek and incorporate feedback from team members and users to refine a solution that meets user needs. [PROGRAM DEVELOPMENT] (P2.3, P1.1)

Development teams that employ user-centered design create solutions (e.g., programs and devices) that can have a large societal impact, such as an app that allows people with speech difficulties to translate hard-to-understand pronunciation into understandable language.

a. Students should begin to seek diverse perspectives throughout the design process to improve their computational artifacts. Considerations of

the end user may include usability, accessibility, age-appropriate content, respectful language, user perspective, pronoun use, color contrast, and ease of use.

2.7 Incorporate existing code, media, and libraries into original programs and give attribution. [PROGRAM DEVELOPMENT] (P4.2, P5.2, P7.3)

Building on the work of others enables students to produce more interesting and powerful creations.

a. Students should use portions of code, algorithms, and/or digital media in their own programs and websites. At this level, they may also import libraries and connect to web application program interfaces (APIs). For example, when creating side-scrolling games, students may incorporate portions of code that create a realistic jump movement from another person's game, and they may also import Creative Commons-lessened images to use in the background.
b. Students should give attribution to the original creator's contributions.

2.8 Systematically test and refine programs using a range of test cases. [PROGRAM DEVELOPMENT] (P6.1)

Test cases are created and analyzed to better meet the needs of users and to evaluate whether programs function as intended. At this level, testing should become a deliberate process that is more iterative, systematic, and proactive than at lower levels.

a. Students will test programs by considering potential errors, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive numbers).

2.9 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. [PROGRAM DEVELOPMENT] (P2.2)

Collaboration is a common and crucial practice in programming development. Often, many individuals and groups work on the interdependent parts of a project together.

a. Students will work collaboratively in groups.
b. Students should assume predefined roles within their teams and manage the project workflow using structured timelines. With teacher guidance, they will begin to create collective goals, expectations, and equitable workloads. For example, students may divide the design stage of a game into planning the storyboard, flowchart, and different parts of the game mechanics. They can then distribute tasks and roles among members of the team and assign deadlines.
c. Students should give attribution to the original creators to acknowledge their contributions.

2.10 Document programs in order to make them easier to follow, test, and debug. [PROGRAM DEVELOPMENT] (P7.2)

Documentation allows creators and others to more easily use and understand a program.

a. Students should provide documentation for end users that explains their artifacts and how they function. For example, students could provide a project overview and clear user instructions.
b. Students should incorporate comments in their product (comments in the code).
c. Students should communicate their process using design documents, flowcharts, and presentations.

IC.2 Impacts of Computing

Conceptual understanding: Computing affects many aspects of the world in both positive and negative ways at local, national, and global levels. Individuals and communities influence computing through their behaviors and cultural and social interactions, and in turn, computing influences new cultural practices. An informed and responsible person should understand the social implications of the digital world, including equity and access to computing.

2.1 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. [CULTURE] (P7.2)

Advancements in computer technology are neither wholly positive nor negative; however, the ways that people use computing technologies have tradeoffs.

a. Students should consider current events related to broad ideas, including privacy, communication, and automation. For example, driverless cars can increase convenience and reduce accidents, but they are also susceptible to hacking. The emerging industry will not only reduce the number of taxi and shared-ride drivers but also create more software engineering and cybersecurity jobs.

2.2 Discuss issues of bias and accessibility in the design of existing technologies. [CULTURE] (P1.2)

a. Students should test and discuss the usability of various technology tools (e.g., apps, games, and devices) with the teacher's guidance. For example, facial recognition software that works better for lighter skin tones was likely developed with a homogeneous testing group and could be improved by sampling a more diverse population. When discussing accessibility, students may notice that allowing a user to change font sizes and colors will not only make an interface usable for people with low vision but also benefits users in various situations, such as in bright daylight or a dark room.

2.3 Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact. [SOCIAL INTERACTIONS] (P2.4, P5.2)

Crowdsourcing is gathering services, ideas, or content from a large group of people, especially from the online community. It can be done at the local level (e.g., classroom or school) or global level (e.g., age-appropriate online communities, like Scratch and Minecraft).

a. Students should collaborate with many contributors. For example, a group of students could combine animations to create a digital community mosaic. They could also solicit feedback from many people through use of online communities and electronic surveys.

2.4 Describe tradeoffs between allowing information to be public and keeping information private and secure. [SAFETY, LAW, & ETHICS] (P7.2)

Sharing information online can help establish, maintain, and strengthen connections between people. For example, it allows artists and designers to display their talents and reach a broad audience; however, security attacks often start with personal information that is publicly available online. Social engineering is based on tricking people into revealing sensitive information, which can be thwarted by being wary of attacks, such as phishing and spoofing.

a. Students should discuss and describe the benefits and dangers of allowing information to be public or kept private and secure.

Level 3A: GRADES 9-10 - Computing Systems

CS.3A Computing Systems

Conceptual understanding: People interact with a wide variety of computing devices that collect, store, analyze, and act upon information in ways that can affect human capabilities both positively and negatively. The physical components (hardware) and instructions (software) that make up a computing system communicate and process information in digital form. An understanding of hardware and software is useful when troubleshooting a computing system that does not work as intended.

3A.1 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. [DEVICES] (P4.1)

Computing devices are often integrated with other systems, including biological, mechanical, and social systems. A medical device can be embedded inside a person to monitor and regulate his or her health, a hearing aid (a type of assistive device) can filter out certain frequencies and magnify others, a monitoring device installed in a motor vehicle can track a person's driving patterns and habits, and a facial recognition device can be integrated into a security system to identify a person. The creation of integrated or embedded systems is not an expectation at this level.

a. Students should be able to identify embedded computer systems.
b. Students should describe the types of data and procedures that are included in the embedded system and explain how the implementation details are hidden from the user. For example, a student might select a car stereo and identify the types of data (radio station presets, station name or number, volume level) and procedures (increase volume, store/recall saved station, mute) it includes.

3A.2 Compare levels of abstraction and interactions between application software, system software, and hardware layers. [HARDWARE & SOFTWARE] (P4.1)

At its most basic level, a computer is composed of physical hardware and electrical impulses. Multiple layers of software are built upon the hardware and interact with the layers above and below them to reduce complexity. System software manages a computing device's resources so that software can interact with hardware. System software is used on many different types of devices, such as smart TVs, assistive devices, virtual components, cloud components, and drones. For example, students may explore the progression from voltage to binary signal to logic gates to adders and so on. Knowledge of specific, advanced terms for computer architecture, such as BIOS, kernel, or bus, is not expected at this level.

a. Students should be able to distinguish between hardware and software.
b. Students should be able to describe the purpose of and differences between system software (i.e., operating system) and application software (i.e., word processor).
c. Students should be able to describe how software and hardware interact. For example, text-editing software interacts with the operating system to receive input from the keyboard, convert the input to bits for storage, and interpret the bits as readable text to display on the monitor.

3A.3 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. [TROUBLESHOOTING] (P6.2)

Troubleshooting complex problems involves the use of multiple sources when researching, evaluating, and implementing potential solutions. Troubleshooting also relies on experience, such as when people recognize that a problem is similar to one they have seen before or adapt solutions that have worked in the past. Examples of complex troubleshooting strategies include resolving connectivity problems, adjusting system configurations and settings, ensuring hardware and software compatibility, and transferring data from one device to another.

a. Students should develop guidelines by creating an artifact that conveys systematic troubleshooting strategies (i.e., create a flow chart or a job aid for a help desk employee).

Level 3A: GRADES 9-10 - Networks and the Internet

NI.3A Networks and the Internet

Conceptual understanding: 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 by providing fast, secure communication and facilitating innovation.

3A.1 Evaluate the scalability and reliability of networks by describing the relationship between routers, switches, servers, topology, and addressing. [NETWORK COMMUNICATION & ORGANIZATION] (P4.1)

Each device is assigned an address that uniquely identifies it on the network. Routers function by comparing IP addresses to determine the pathways packets should take to reach their destination. Switches function by comparing MAC addresses to determine which computers or network segments will receive frames. Students could use online network simulators to experiment with these factors.

a. Students should be able to define a MAC address – what it is and how it is used.
b. Students should be able to explain what a router and a switch are and how they work inside a network.
c. Students should be able to define what a server is and how it is used in a network.
d. Students should be able to list various types of network topology and explain why each is used.
e. Students should be able to verbally and visually explain how addressing, routers, switches, and servers all work together in a network.

3A.2 Give examples to illustrate how sensitive data can be affected by malware and other attacks. [CYBERSECURITY] (P7.2)

Network security depends on a combination of hardware, software, and practices that control access to data and systems. The needs of users and the sensitivity of data determine the level of security implemented. Potential security problems, such as denial-of-service attacks, ransomware, viruses, worms, spyware, and phishing, present threats to sensitive data.

a. Students should be able to discuss how sensitive data can be affected by malware and other attacks. Students might reflect on case studies or current events in which governments or organizations experienced data leaks or data loss as a result of these types of attacks.

3A.3 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. [CYBERSECURITY] (P3.1, 3.3)

Security measures may include physical security tokens, two-factor authentication, and biometric verification. Potential security problems, such as denial-of-service attacks, ransomware, viruses, worms, spyware, and phishing, exemplify why sensitive data should be securely stored and transmitted. The timely and reliable access to data and information services by authorized users, referred to as availability, is ensured through adequate bandwidth, backups, and other measures.

a. Students should understand the different types of security problems and the different types of devices that can be impacted. Potential security problems may include issues such as denial-of-service attacks, ransomware, viruses, worms, spyware, phishing, and social engineering. Some types of devices impacted may include laptops, tablets, cell phones, self-driving cars, ATMs, and others.
b. Students should systematically evaluate different security measures based on efficiency, feasibility, and ethical impacts. Students might address issues such as how efficiency affects feasibility or whether a proposed approach raises ethical concerns.

3A.4 Compare various security measures considering tradeoffs between the usability and security of a computing system. [CYBERSECURITY] (P6.3)

Security measures may include physical security tokens, two-factor authentication, and biometric verification, but choosing security measures involves tradeoffs between the usability and security of the system. The needs of users and the sensitivity of data determine the level of security implemented.

a. Students should be able to explain different types of security measures and discuss the tradeoffs between usability and security. For example, students might discuss computer security policies in place at the local level that present a tradeoff between usability and security, such as a web filter that prevents access to many educational sites but keeps the campus network safe.

3A.5 Explain tradeoffs when selecting and implementing cybersecurity recommendations. [CYBERSECURITY] (P7.2)

Network security depends on a combination of hardware, software, and practices that control access to data and systems. The needs of users and the sensitivity of data determine the level of security implemented. Every security measure involves tradeoffs between the accessibility and security of the system.

a. Students should be able to describe, justify, and document choices they make using terminology appropriate for the intended audience and purpose. Students could debate issues from the perspective of diverse audiences, including individuals, corporations, privacy advocates, security experts, and government.

Level 3A: GRADES 9-10 - Data and Analysis

DA.3A Data and Analysis ()

Conceptual understanding: Computing systems exist to process data. The amount of digital data generated in the world is rapidly expanding, so the need to process data effectively is increasingly important. Data is collected and stored so that it can be analyzed to better understand the world and make more accurate predictions.

3A.1 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. [STORAGE] (P4.1)

a. Students should be able to translate between different bit representations. For example, convert hexadecimal color codes to decimal percentages, ASCII/Unicode representation, or converting binary to base 10.
b. Students should be able to discuss how data sequences can be interpreted in a variety of formats. For example, text, numbers, sound, and images.

3A.2 Evaluate the tradeoffs in how data elements are organized and where data is stored. [STORAGE] (P3.3)

People make choices about how data elements are organized and where data is stored. These choices affect cost, speed, reliability, accessibility, privacy, and integrity.

a. Students should evaluate whether a chosen solution is most appropriate for a particular problem. Students might consider the cost, speed, reliability, accessibility, privacy, and integrity tradeoffs between storing photo data on a mobile device versus in the cloud.

3A.3 Collect, transform, and organize data to help others better understand a problem. [COLLECTION, VISUALIZATION, & TRANSFORMATION] (P4.4)

People transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information. Examples include visualization, aggregation, rearrangement, and application of mathematical operations. People use software tools or programming to create powerful, interactive data visualizations and perform a range of mathematical operations to transform and analyze data.

a. Students should use various data collection techniques for different types of computational problems. For example, user surveys, mobile device GPS, social media data sets, etc.
b. Use computational tools to collect, transform, and organize data to help others better understand a problem.
c. Students should use data analysis to identify significant patterns in data sets.

3A.4 Create and evaluate computational models that represent real-world systems. [INFERENCE & MODELS] (P4.4)

Computational models make predictions about processes or phenomena based on selected data and features. The amount, quality, and diversity of data and the features chosen can affect the quality of a model and ability to understand a system. Predictions or inferences are tested to validate models.

a. Students should create computational models that simulate real-world systems (e.g., ecosystems, epidemics, spread of disease).
b. Students should analyze and evaluate the ability of models and simulations to formulate, refine, and test hypotheses.

Level 3A: GRADES 9-10 - Algorithms and Programming

AP.3A Algorithms and Programming

Conceptual understanding: An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are translated into programs, or code, to provide instructions for computing devices. Algorithms and programming control all computing systems, empowering people to communicate with the world in new ways and solve compelling problems. The development process to

create meaningful and efficient programs involves choosing which information to use and how to process and store it, breaking apart large problems into smaller ones, recombining existing solutions, and analyzing different solutions.

3A.1 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. [ALGORITHMS] (P5.2)

A prototype is a computational artifact that demonstrates the core functionality of a product or process. Prototypes are useful for getting early feedback in the design process and can yield insight into the feasibility of a product. The process of developing computational artifacts embraces both creative expression and the exploration of ideas to create prototypes and solve computational problems.

a. Students create artifacts that are personally relevant or beneficial to their community and beyond. Students should develop artifacts in response to a task or a computational problem that demonstrate the performance, reusability, and ease of implementation of an algorithm.

3A.2 Use lists and functions to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. [VARIABLES] (P4.1)

a. Students should be able to identify common features in multiple segments of code and substitute a single segment that uses lists (arrays) or functions to account for the differences.

3A.3 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. [CONTROL] (P5.2)

Implementation includes the choice of programming language, which affects the time and effort required to create a program. Readability refers to how clear the program is to other programmers and can be improved through documentation. The discussion of performance is limited to a theoretical understanding of execution time and storage requirements; a quantitative analysis is not expected. Control structures at this level may include conditional statements, loops, event handlers, and recursion.

a. Students should be able to justify by explaining the benefits and drawbacks of the selection of specific control structures with regard to implementation, readability, and program performance. For example, students might compare the readability and program performance of iterative and recursive implementations of procedures that calculate the Fibonacci sequence.

3A.4 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. [CONTROL] (P5.2)

In this context, relevant computational artifacts include programs, mobile apps, or Web apps. Events can be user-initiated, such as a button press, or system-initiated, such as a timer firing. At previous levels, students have learned to create and call procedures. Here, students design procedures that are called by events.

a. Students will design procedures that are called by events. Students might create a mobile app that updates a list of nearby points of interest when the device detects that its location has been changed.

3A.5 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. [MODULARITY] (P3.2)

a. Students should decompose complex problems into manageable subproblems that could potentially be solved with programs or procedures that already exist. For example, students could create an app to solve a community problem by connecting to an online database through an application programming interface (API).

3A.6 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. [MODULARITY] (P5.2)

Computational artifacts can be created by combining and modifying existing artifacts or by developing new artifacts. Examples of computational artifacts include programs, simulations, visualizations, digital animations, robotic systems, and apps. Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks. The focus at this level is understanding a program as a system with relationships between modules.

a. Students will create artifacts by using procedures within a program, combinations of data, and procedures, or independent but interrelated programs. The choice of implementation, such as programming language or paradigm, may vary. Students could incorporate computer vision libraries to increase the capabilities of a robot or leverage open-source JavaScript libraries to expand the functionality of a Web application.

3A.7 Systematically design and develop programs for broad audiences by incorporating feedback from users. [PROGRAM DEVELOPMENT] (P5.1)

Examples of programs could include games, utilities, and mobile applications. Students at lower levels collect feedback and revise programs.

a. Students should do so through a systematic process that includes feedback from broad audiences. Students might create a user satisfaction survey and brainstorm distribution methods that could yield feedback from a diverse audience, documenting the process they took to incorporate selected feedback in product revisions.

3A.8 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. [PROGRAM DEVELOPMENT] (P7.3)

Examples of software licenses include copyright, freeware, and many open-source licensing schemes. At previous levels, students adhered to licensing schemes.

a. Students should consider licensing implications for their own work, especially when incorporating libraries and other resources. Students might consider two software libraries that address a similar need, justifying their choice based on the library that has the least restrictive license.

3A.9 Evaluate and refine computational artifacts to make them more usable and accessible. [PROGRAM DEVELOPMENT] (P6.3)

Testing and refinement is the deliberate and iterative process of improving a computational artifact. This process includes debugging (identifying and fixing errors) and comparing actual outcomes to intended outcomes.

a. Students should respond to the changing needs and expectations of end users and improve the performance, reliability, usability, and accessibility of artifacts. For example, students could incorporate feedback from a variety of end users to help guide the size and placement of menus and buttons in a user interface.

3A.10 Design and develop computational artifacts working in team roles using collaborative tools. [PROGRAM DEVELOPMENT] (P2.4)

Collaborative tools could be as complex as a source code version control system or as simple as a collaborative word processor. Team roles in pair programming are driver and navigator but could be more specialized in larger teams. As programs grow more complex, the choice of resources that aid program development becomes increasingly important and should be made by the students.

a. Students will work in teams using collaborative tools to design and develop computational artifacts. Students might work as a team to develop a mobile application that addresses a problem relevant to the school or community, selecting appropriate tools to establish and manage the project timeline; design, share, and revise graphical user interface elements; and track planned, in-progress, and completed components.

3A.11 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. [PROGRAM DEVELOPMENT] (P7.2)

Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. These modules can be procedures within a program; combinations of data and procedures; or independent, but interrelated, programs. The development of complex programs is aided by resources such as libraries and tools to edit and manage parts of the program.

a. Students will document design decisions using text, graphics, presentations, and/or demonstrations.

Level 3A: GRADES 9-10 - Impacts of Computing

IC.3A Impacts of Computing

Conceptual understanding: Computing affects many aspects of the world in both positive and negative ways at local, national, and global levels. Individuals and communities influence computing through their behaviors and cultural and social interactions, and in turn, computing influences new cultural practices. An informed and responsible person should understand the social implications of the digital world, including equity and access to computing.

3A.1 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. [CULTURE] (P1.2)

Computing may improve, harm, or maintain practices. Equity deficits, such as minimal exposure to computing, access to education, and training opportunities, are related to larger, systemic problems in society.

a. Students should be able to evaluate the accessibility of a product to a broad group of end users, such as people who lack access to broadband or who have various disabilities.
b. Students should also begin to identify potential bias during the design process to maximize accessibility in product design.

3A.2 Test and refine computational artifacts to reduce bias and equity deficits. [CULTURE] (P1.2)

Biases could include incorrect assumptions developers have made about their user base. Equity deficits include minimal exposure to computing, access to education, and training opportunities.

a. Students should begin to identify potential bias during the design process to maximize accessibility in product design and become aware of professionally accepted accessibility standards to evaluate computational artifacts for accessibility.

3A.3 Demonstrate ways a given algorithm applies to problems across disciplines. [CULTURE] (P3.1)

Computation can share features with disciplines, such as art and music, by algorithmically translating human intention into an artifact.

a. Students should be able to identify real-world problems that span multiple disciplines, such as increasing bike safety with new helmet technology, and that can be solved computationally.

3A.4 Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. [SOCIAL INTERACTIONS] (P2.4)

Many aspects of society, especially careers, have been affected by the degree of communication afforded by computing. The increased connectivity between people in different cultures and in different career fields has changed the nature and content of many careers.

a. Students should explore different collaborative tools and methods used to solicit input from team members, classmates, and others, such as participation in online forums or local communities. For example, students could compare ways different social media tools could help a team become more cohesive

3A.5 Explain the beneficial and harmful effects that intellectual property laws can have on innovation. [SAFETY, LAW, & ETHICS] (P7.3)

Laws govern many aspects of computing, such as privacy, data, property, information, and identity. These laws can have beneficial and harmful effects, such as expediting or delaying advancements in computing and protecting or infringing upon people's rights. International differences in laws and ethics have implications for computing. For examples, laws that mandate the blocking of some file-sharing websites may reduce online piracy but can restrict the right to access information. Firewalls can be used to block harmful viruses and malware but can also be used for media censorship.

a. Students should be aware of intellectual property laws and be able to explain how they are used to protect the interests of innovators and how patent trolls abuse the laws for financial gain.

3A.6 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. [SAFETY, LAW, & ETHICS] (P7.2)

Data can be collected and aggregated across millions of people, even when they are not actively engaging with or physically near the data collection devices. This

automated and non-evident collection can raise privacy concerns, such as social media sites mining an account even when the user is not online. Other examples include surveillance video used in a store to track customers for security or information about purchase habits or the monitoring of road traffic to change signals in real time to improve road efficiency without drivers being aware. Methods and devices for collecting data can differ by the amount of storage required, the level of detail collected, and sampling rates.

a. Students should be able to explain the privacy concerns related to the collection and generation of data through automated processes.

3A.7 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics. [SAFETY, LAW, & ETHICS] (P7.3)

Laws govern many aspects of computing, such as privacy, data, property, information, and identity. International differences in laws and ethics have implications for computing.

a. Students should evaluate the social and economic implications of privacy in the context of safety, law, or ethics. For example, students might review case studies or current events that present an ethical dilemma when an individual's right to privacy is at odds with the safety, security, or well-being of a community.

Appendix D: 21st Century Learning

	Unit	1	2	3	4	5	6	7	8	9
Standards										
CS		X	X	X			X			
4C		X	X	X	X	X		X		
IMTS		X			X	X	X	X	X	X
LCS			X	X			X		X	

21st Century Learning - Framework Elements

CS **Core Subjects and 21st Century Themes**:
Incorporation in Units: Units that emphasize the importance of core academic subjects (like digital literacy in Unit 1) and integrate 21st-century themes such as global awareness, civic literacy, and environmental literacy.

4C **Learning and Innovation Skills (4Cs)**:
Critical thinking, Communication, Collaboration, and Creativity: Units like Unit 3 (21st Century Toolbox) and Unit 5 (Block-Based Programming) heavily focus on these skills, as they require students to collaborate, think critically, and communicate their ideas effectively.

IMTS **Information, Media, and Technology Skills**:
Application in Units: Units focused on digital design, cybersecurity, and data analysis (Units 4, 6, 7, 8, and 9) emphasize the ability to effectively use technology and manage information.

LCS **Life and Career Skills**:
Integration: These skills, such as leadership, initiative, and flexibility, are core to Units 2 (Student Organizations and Leadership) and Unit 8 (App Design).