



# Willingboro Public Schools

---

*“Where Excellence is the Expectation”*

## **Willingboro Public Schools Computer Science I Curriculum**

**Revised April 2025  
Jennifer Brandon**

**Course Sequence/Table of Contents:**

|          |  |
|----------|--|
| <b>1</b> | <b>Unit 1: Foundations of Computer Science: From Data to Decisions</b>         |
| <b>2</b> | <b>Unit 2: Coding with Rhythm – Loops and Music Programming</b>                |
| <b>3</b> | <b>Unit 3: Programming Structures – Graphics, Functions, and Lists</b>         |
| <b>4</b> | <b>Unit 4: Code in Action – Music, Web, and Careers in Computer Science</b>    |
| <b>7</b> | <b>Appendix A: Instructional Best Practices and Exemplars</b>                  |
| <b>8</b> | <b>Appendix B: Exemplars and Explanations</b>                                  |
| <b>9</b> | <b>Appendix C: Classroom Philosophy, Schedule, Structure, and Expectations</b> |

**[Click here for the Computer Science I 24-25 Pacing Guide](#)**

| Overview  | Content Standards   | Unit Focus   |
|---|---|--|
| <b>Unit 1<br/>Foundations of<br/>Computer<br/>Science: From<br/>Data to<br/>Decisions</b> | <ul style="list-style-type: none"> <li>8.1.12.AP.1. Design algorithms to solve computational problems using a combination of original and existing algorithms.</li> <li>8.1.12.AP.2. Create generalized computational solutions using collections instead of repeatedly using simple variables</li> </ul> | Students will explore the foundational principles of computer science, including the history of computing, hardware and software, programming with Python, data representation, and decision-making through algorithms. The unit emphasizes how computers process data and execute logical decisions to solve real-world problems. |

| Overview   | Content Standards   | Unit Focus   |
|--|---|--|
| <i>Unit 1:<br/>Suggested Open Educational Resources</i>                        | Project STEM CS Python Fundamentals   |  |
| <b>Unit 2<br/>Coding with Rhythm – Loops and Music Programming</b>             | <ul style="list-style-type: none"> <li>● 8.1.12.AP.1. Design algorithms to solve computational problems using a combination of original and existing algorithms.</li> <li>● 8.1.12.AP.2. Create generalized computational solutions using collections instead of repeatedly using simple variables</li> </ul> | Students will explore the power of repetition in coding by learning how loops function in programming and how these same principles can be applied to music creation. Using Python and EarSketch, students will design programs that incorporate loops, algorithms, and musical elements, blending computational thinking with creativity.   |
| <i>Unit 2:<br/>Suggested Open Educational Resources</i>                        | Project STEM CS Python Fundamentals   |  |
| <b>Unit 3<br/>Programming Structures – Graphics, Functions, and Lists</b>      | <ul style="list-style-type: none"> <li>● 8.1.12.AP.1. Design algorithms to solve computational problems using a combination of original and existing algorithms.</li> <li>● 8.1.12.AP.2. Create generalized computational solutions using collections instead of repeatedly using simple variables</li> </ul> | Students will develop an understanding of key programming structures—graphics, functions, and lists—by creating visual outputs, organizing code into reusable components, and managing collections of data. Through hands-on projects, students will see how these structures allow programmers to design, organize, and optimize their work for both efficiency and creativity.         |
| <i>Unit 3:<br/>Suggested Open Educational Resources</i>                        | Project STEM CS Python Fundamentals   |  |
| <b>Unit 4<br/>Code in Action – Music, Web, and Careers in Computer Science</b> | <ul style="list-style-type: none"> <li>● 8.1.12.AP.1. Design algorithms to solve computational problems using a combination of original and existing algorithms.</li> <li>● 8.1.12.AP.2. Create generalized computational solutions using collections instead of repeatedly using simple variables</li> </ul> | Students will apply advanced programming structures, explore music programming, investigate how the internet works, and examine career pathways in computer science. By working with 2D lists, music coding in EarSketch, and web development with HTML, students will see coding in action across multiple domains and connect their skills to real-world applications and professions. |
| <i>Unit 4:<br/>Suggested Open Educational Resources</i>                        | Project STEM CS Python Fundamentals   |  |

**Unit 1: Foundations of Computer Science: From Data to Decisions**

**Overview**

This unit introduces students to the field of computer science as both a discipline and a problem-solving tool. Students will begin with a study of computing history and core concepts such as binary, hardware/software systems, and the distinction between analog and digital. They will learn Python as their first programming language, writing simple programs that use input/output, variables, and calculations. Students will also explore how computers represent and manipulate data, from small-scale calculations to the concept of “big data.” Finally, the unit transitions into decision-making using functions, conditional logic, and algorithms, culminating in projects such as creating a chatbot and analyzing real data sets.

**Essential Questions**

- What is computer science, and why is it important in today’s world?
- How do computers process, represent, and store data?
- How can programming languages like Python be used to solve problems and communicate with computers?
- What role do algorithms and logical decision-making play in computer science?
- How has the history of computing shaped the tools and technologies we use today?

**Enduring Understandings**

- Computer science is the study of problem solving with computers, involving both hardware systems and software design.
- Computers rely on data representation systems, such as binary, to process information.
- Programming languages allow humans to communicate with computers through precise instructions.
- Logical decision-making and algorithms form the backbone of computational problem-solving.
- The history of computing provides context for current technologies and their impact on society.

**Unit 1 Create Drawings**

**Content Standards**

- 8.1.12.AP.1. Design algorithms to solve computational problems using a combination of original and existing algorithms.
- 8.1.12.AP.2. Create generalized computational solutions using collections instead of repeatedly using simple variables.

**Unit 1 Create Drawings**

**Core Ideas**

- Computer science is the study of how computers solve problems through hardware, software, and programming.
- Data can be represented in different forms, including binary, analog, and digital, to make information usable by computers.
- Programming languages like Python provide tools for creating instructions, solving problems, and automating tasks.
- Algorithms and decision-making structures (such as conditionals) are essential for creating efficient and effective solutions.
- Historical developments in computing shape modern technology and its applications in everyday life.
- Data analysis, from simple calculations to “big data,” informs decisions across disciplines.

**Student Learning Objectives**

Students will be able to...

- Define computer science and explain its relevance across industries and society.
- Identify the roles of hardware and software in a computer system.
- Write and run simple Python programs using input, output, variables, and operators.
- Differentiate between analog and digital systems and explain binary representation.
- Perform number calculations, modular division, and apply built-in Python functions.
- Analyze and manipulate data sets, recognizing the implications of “big data.”
- Use functions and conditionals to design programs that make decisions.
- Apply algorithmic thinking to solve problems and create a functional chatbot.

**Integrated Accommodations and Modifications**

| Special Education Students  | English Language Learners   | At Risk   |
|---|---|---|
| <ul style="list-style-type: none"> <li>● Utilize modifications &amp; accommodations delineated in the student’s IEP</li> <li>● Provide additional manipulatives to support instruction</li> <li>● Allow for alternative strategies to solve algorithms or tasks</li> <li>● Provide the steps needed to complete the task</li> <li>● Model frequently</li> <li>● Provide repetition and practice.</li> <li>● Use visuals to demonstrate/model the processes</li> <li>● Restate, reread, and clarify directions/questions</li> <li>● Ask students to restate information, directions, and assignments.</li> <li>● Provide copy of class notes</li> <li>● Distribute study guide for classroom tests.</li> <li>● Provide preferential seating to be mutually determined by the student and teacher</li> <li>● Provide extra textbooks for home.</li> <li>● Provide regular parent/ school communication</li> </ul> | <p>WIDA Can Do Descriptors<br/> <a href="https://wida.wisc.edu/teach/can-do/descriptors">https://wida.wisc.edu/teach/can-do/descriptors</a></p> <ul style="list-style-type: none"> <li>● Modify Assignments</li> <li>● Use testing and portfolio assessment</li> <li>● Utilize Native Language Translation (peer, online assistive technology, translation device, bilingual dictionary)</li> <li>● Repeat, rephrase, paraphrase key concepts and directions</li> <li>● Allow for extended time for assignment completion as needed</li> <li>● Highlight key vocabulary</li> <li>● Define essential vocabulary in context</li> <li>● Use graphic organizers, visuals, manipulatives and other concrete materials</li> <li>● Use gestures, facial expressions and body language</li> <li>● Read aloud</li> <li>● Build on what students already know and prior experience</li> </ul> | <ul style="list-style-type: none"> <li>● Pair visual prompts with verbal presentations</li> <li>● Ask students to restate information, directions, and assignments.</li> <li>● Provide repetition and and practice</li> <li>● Model skills / techniques to be mastered.</li> <li>● Provide extended time to complete class work</li> <li>● Provide copy of class notes</li> <li>● Provide preferential seating to be mutually determined by the student and teacher</li> <li>● Allow the use of a computer to complete assignments.</li> <li>● Establish expectations for correct spelling on assignments</li> <li>● Provide extra textbooks for home.</li> <li>● Provide Peer Support</li> <li>● Increase one on one time</li> </ul> |

|  |  |                 |
|--|--|-----------------|
| <ul style="list-style-type: none"> <li>● Allow extended time to complete assignment</li> <li>● Establish procedures for accommodations / modifications for assessments</li> <li>● Allow student to take/complete tests in an alternate setting as needed</li> </ul> <p><u>Appendix A: Special Education Accommodations and Modifications</u></p>   |  |                 |
| <b>Gifted and Talented Students</b>  |  | <b>504 Plan</b> |
| <ul style="list-style-type: none"> <li>● Utilize advanced, accelerated, or compacted content</li> <li>● Provide assignments that emphasize higher- level thinking skills.</li> <li>● Allow for individual student interest</li> <li>● Gear assignments to development in areas of affect, creativity, cognition, and research skills</li> <li>● Allow for a variety in types of resources</li> <li>● Provide problem-based assignments with planned scope and sequence</li> <li>● Utilize inquiry-based instruction</li> <li>● Adjust the pace of lessons</li> <li>● Utilize Choice Boards</li> <li>● Provide Problem-Based Learning</li> <li>● Establish flexible Grouping</li> </ul>   | <ul style="list-style-type: none"> <li>● Pair visual prompts with verbal presentations</li> <li>● Ask students to restate information, directions, and assignments.</li> <li>● Provide repetition and and practice</li> <li>● Model skills / techniques to be mastered.</li> <li>● Provide extended time to complete class work</li> <li>● Provide copy of class notes</li> <li>● Break long assignments into smaller parts</li> <li>● Assist student in setting short term goals</li> <li>● Allow for preferential seating to be mutually determined by the student and teacher</li> <li>● Provide extra textbooks for home.</li> <li>● Model and reinforce organizational systems (i.e. color-coding)</li> <li>● Write out homework assignments, check student's recording of assignments</li> </ul> |                 |
| <b>Interdisciplinary Connections</b>   |  |                 |
| <p><b>ELA Standards</b></p> <ul style="list-style-type: none"> <li>● W.9-10.9. Draw evidence from literary or nonfiction informational texts to support analysis, reflection, and research.</li> <li>● SL.9-10.1. Initiate and participate effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with peers on grades 9–10 topics, texts, and issues, building on others’ ideas and expressing their own clearly and persuasively.</li> <li>● SL.9-10.2. Integrate multiple sources of information presented in diverse media or formats (e.g., visually, quantitatively, qualitatively, orally) evaluating the credibility and accuracy of each source.</li> <li>● W.11-12.9. Draw evidence from literary or informational texts to support analysis, reflection, and research.</li> <li>● SL.11-12.1. Initiate and participate effectively in a range of collaborative discussions (one-on- one, in groups, and teacher-led) with peers on grades 11–12 topics, texts, and issues, building on others’ ideas and expressing their own clearly and persuasively.</li> <li>● SL.11-12.2. Integrate multiple sources of information presented in diverse media or formats (e.g., visually, quantitatively, qualitatively, orally) evaluating the credibility and accuracy of each source.</li> </ul> |  |                 |
| <b>Career Readiness, Life Literacies and Key Skills</b>  |  |                 |

**Career Readiness, Life Literacies and Key Skills Practices**

- Demonstrate creativity and innovation.
- Utilize critical thinking to make sense of problems and persevere in solving them.
- Model integrity, ethical leadership and effective management.
- Plan education and career paths aligned to personal goals.
- Use technology to enhance productivity, increase collaboration and communicate effectively.
- Work productively in teams while using cultural/global competence.

**Career Readiness, Life Literacies and Key Skills Standards**

**9.4 Life Literacies and Key Skills**

21st Century Skills

- 9.4.12.CI.1. Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2. Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3. Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1. Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
- 9.4.12.CT.2. Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).

Technology Integration

- 9.4.12.IML.4. Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).

**Climate Change**

**Addressed in Unit 2**

**SEL Competencies**

- **Self - Awareness**
- **Self - Management**
- **Social Awareness**
- **Responsible Decision Making**
- **Relationship Skills**

<https://www.nj.gov/education/safety/wellness/selearning/index.shtml>

**District/School Formative Assessment Plan**

*Formative assessment informs instruction and is ongoing throughout a unit to determine how students are progressing against the standards.*

Teachers are encouraged to incorporate Formative Assessments into all lessons. During instruction, teachers will collect ongoing information on students' mastery of content through a variety of methods:

- Questioning: using Socratic method, probing questions, a hierarchical system in complexity (Bloom's Taxonomy)

**District/School Summative Assessment Plan**

*Summative assessment is an opportunity for students to demonstrate mastery of the skills taught during a particular unit.*

**Summative Assessment:**

- **Silly Sentences Program:** Write a Python program that takes user input and generates silly sentences, demonstrating understanding of input, output, variables, and data types.

|   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• Exit tickets, rotational activities (stations), quizzes, and small group activities</li> <li>• Classwork, homework, group work (formative assessment)</li> <li>• Pre-Assessment, teacher’s observation, class discussion, and journal</li> </ul> | <ul style="list-style-type: none"> <li>• <b>Hardware &amp; Software Diagram:</b> Create a labeled diagram or infographic explaining the relationship between hardware and software.</li> <li>• <b>Binary &amp; Data Representation Quiz/Project:</b> Students demonstrate how to convert numbers and text to binary and explain analog vs. digital differences.</li> <li>• <b>First Program Showcase:</b> Present and explain their first working Python program, including the steps taken to build it.</li> <li>• <b>Room Area Calculator:</b> Create a Python program that calculates the area of a room, integrating input, calculations, and output.</li> <li>• <b>History of Computing Timeline:</b> Develop a digital timeline or presentation showing key events and figures in computer science history.</li> <li>• <b>Big Data Case Study:</b> Analyze an example of “big data” in the real world and present findings about its uses and challenges.</li> <li>• <b>Function Application Project:</b> Use built-in Python functions to solve a real-world style problem (e.g., random number simulation).</li> <li>• <b>Chatbot Project:</b> Build a basic chatbot that uses conditionals (if, else, else-if) and logical operators to carry on a simple conversation.</li> <li>• <b>Algorithm Challenge:</b> Write and trace algorithms to solve logic-based problems, then compare efficiency.</li> <li>• <b>Decision Tree Diagram:</b> Create a visual representation of how conditional statements (if/else) control program flow.</li> <li>• <b>Function-Based Mini Project:</b> Write a program that defines and uses multiple functions, applying parameters and return values.</li> </ul> <p><b>Other Summative Assessments:</b> Teachers are encouraged to design and their own assessments (topic/module tests and quizzes) individually and/or with their department or grade-level partners, as per Uniform Grading Profile.</p> |
|---|--|

**Targeted Academic Vocabulary**

computer science, Python, program, hardware, software, input, output, variable, data type, analog, digital, binary, calculation, modular division, built-in function, random number, big data, dataset, function, conditional, if statement, else, else-if, logical operator, algorithm, algorithmic thinking, chatbot

| District/School Tasks   | District/School Primary and Supplementary Resources   |
|---|---|
| <ul style="list-style-type: none"> <li>• See Formative/Summative Assessments</li> </ul> | <p><b>District-Mandated Resources</b></p> <ul style="list-style-type: none"> <li>• Project STEM CS Python Fundamentals</li> </ul> |

|   |  |
|---|--|
|   | <p><b>Other Resources:</b></p> <ul style="list-style-type: none"> <li>• <a href="#">Teacher Handbook</a></li> <li>• <a href="#">Creative Task Examples</a></li> <li>• <a href="#">Resources from Amazon Future Engineer</a></li> <li>• <a href="#">The First Programming Languages (Diversity, Equity and Inclusion)</a></li> <li>• <a href="#">Learn Python</a></li> <li>• <a href="#">When Women Stopped Coding-CommonLit (Diversity, Equity and Inclusion)</a></li> </ul> |
| <b>Instructional Best Practices and Exemplars</b> |  |
| See Appendix B                                    |  |
| <b>Pacing Guide</b>                               |  |
| <b>Computer Science I Pacing Guide</b>            |  |

| <b>Unit 2: Coding with Rhythm – Loops and Music Programming</b>   |   |
|---|---|
| <b>Overview</b>   |   |
| <p>This unit introduces students to the concept of loops as a fundamental programming structure, showing how repetition and iteration make code more efficient and powerful. Students will learn about count variables, ways to end loops, and applications of range and for-loops. They will revisit data concepts and connect looping to algorithms, tracing, and simulations. Building on these skills, students will transition to EarSketch, where they will apply coding to compose and manipulate music. Students will explore sound effects, tempo, pitch, and musical form while also learning about debugging, documentation, and copyright. Through hands-on projects, such as designing a ringtone or beat, students will experience how coding can be used for both technical problem-solving and artistic expression.</p> |   |
| <b>Essential Questions</b>  | <b>Enduring Understandings</b>  |
| <ul style="list-style-type: none"> <li>• Why are loops an essential part of programming?</li> <li>• How do algorithms and simulations help us model and solve real-world problems?</li> <li>• In what ways do repetition and patterns connect coding and music?</li> </ul>  | <ul style="list-style-type: none"> <li>• Loops allow programmers to efficiently repeat instructions and reduce redundancy in code.</li> <li>• Algorithms, supported by looping structures, provide systematic solutions to problems.</li> </ul> |

|   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• How can programming be used as a tool for creative expression?</li> <li>• Why is debugging, documentation, and ethical awareness important in computer science?</li> </ul> | <ul style="list-style-type: none"> <li>• Music and programming both rely on structure, rhythm, and repetition to create meaning.</li> <li>• Creative applications of coding, such as music programming, expand how computer science is used in the real world.</li> <li>• Debugging, documentation, and ethical practices ensure that programs are functional, understandable, and responsible.</li> </ul> |
|---|--|

**Unit 2: Coding with Rhythm – Loops and Music Programming**

|   |
|---|
| <b>Core Ideas</b>   |
| <ul style="list-style-type: none"> <li>• Loops make programming more efficient by allowing repetition of actions with fewer instructions.</li> <li>• Algorithms and simulations rely on structured repetition to model real-world processes.</li> <li>• Music, like code, is built on patterns and repetition, making it a natural context for programming.</li> <li>• Programming tools like EarSketch allow students to combine creativity and computer science.</li> <li>• Debugging, documenting, and evaluating code are essential for accuracy and clarity.</li> <li>• Copyright and ethics are important considerations when creating with digital media.</li> </ul> |

**Student Learning Objectives**

|   |
|---|
| <p>Students will be able to...</p> <ul style="list-style-type: none"> <li>• Define and apply loops, count variables, and range functions in Python programs.</li> <li>• Use for-loops and conditional logic to design efficient algorithms.</li> <li>• Apply algorithm tracing and simulation to solve mathematical and computational challenges.</li> <li>• Create music compositions in EarSketch using loops, effects, tempo, pitch, and custom functions.</li> <li>• Debug and document code to ensure correctness and readability.</li> <li>• Explain the importance of copyright and ethical considerations in digital creation.</li> <li>• Demonstrate how programming concepts can be applied in creative contexts by designing a ringtone or musical project.</li> </ul> |
|---|

**Integrated Accommodations and Modifications**

| Special Education Students  | English Language Learners   | At Risk  |
|---|---|--|
| <ul style="list-style-type: none"> <li>• Utilize modifications &amp; accommodations delineated in the student’s IEP</li> <li>• Provide additional manipulatives to support instruction</li> <li>• Allow for alternative strategies to solve algorithms or tasks</li> <li>• Provide the steps needed to complete the task</li> </ul> | <p>WIDA Can Do Descriptors<br/> <a href="https://wida.wisc.edu/teach/can-do/descriptors">https://wida.wisc.edu/teach/can-do/descriptors</a></p> <ul style="list-style-type: none"> <li>• Modify Assignments</li> <li>• Use testing and portfolio assessment</li> <li>• Utilize Native Language Translation (peer, online assistive technology, translation device, bilingual dictionary)</li> </ul> | <ul style="list-style-type: none"> <li>• Pair visual prompts with verbal presentations</li> <li>• Ask students to restate information, directions, and assignments.</li> <li>• Provide repetition and practice</li> <li>• Model skills / techniques to be mastered.</li> <li>• Provide extended time to complete class work</li> </ul> |

|  |   |   |
|--|---|---|
| <ul style="list-style-type: none"> <li>● Model frequently</li> <li>● Provide repetition and practice.</li> <li>● Use visuals to demonstrate/model the processes</li> <li>● Restate, reread, and clarify directions/questions</li> <li>● Ask students to restate information, directions, and assignments.</li> <li>● Provide copy of class notes</li> <li>● Distribute study guide for classroom tests.</li> <li>● Provide preferential seating to be mutually determined by the student and teacher</li> <li>● Provide extra textbooks for home.</li> <li>● Provide regular parent/ school communication</li> <li>● Allow extended time to complete assignment</li> <li>● Establish procedures for accommodations / modifications for assessments</li> <li>● Allow student to take/complete tests in an alternate setting as needed</li> </ul> <p><u>Appendix A: Special Education Accommodations and Modifications</u></p> | <ul style="list-style-type: none"> <li>● Repeat, rephrase, paraphrase key concepts and directions</li> <li>● Allow for extended time for assignment completion as needed</li> <li>● Highlight key vocabulary</li> <li>● Define essential vocabulary in context</li> <li>● Use graphic organizers, visuals, manipulatives and other concrete materials</li> <li>● Use gestures, facial expressions and body language</li> <li>● Read aloud</li> <li>● Build on what students already know and prior experience</li> </ul>  | <ul style="list-style-type: none"> <li>● Provide copy of class notes</li> <li>● Provide preferential seating to be mutually determined by the student and teacher</li> <li>● Allow the use of a computer to complete assignments.</li> <li>● Establish expectations for correct spelling on assignments</li> <li>● Provide extra textbooks for home.</li> <li>● Provide Peer Support</li> <li>● Increase one on one time</li> </ul> |
| <p style="text-align: center;"><b>Gifted and Talented Students</b></p> <ul style="list-style-type: none"> <li>● Utilize advanced, accelerated, or compacted content</li> <li>● Provide assignments that emphasize higher- level thinking skills.</li> <li>● Allow for individual student interest</li> <li>● Gear assignments to development in areas of affect, creativity, cognition, and research skills</li> <li>● Allow for a variety in types of resources</li> <li>● Provide problem-based assignments with planned scope and sequence</li> <li>● Utilize inquiry-based instruction</li> <li>● Adjust the pace of lessons</li> <li>● Utilize Choice Boards</li> <li>● Provide Problem-Based Learning</li> <li>● Establish flexible Grouping</li> </ul>  | <p style="text-align: center;"><b>504 Plan</b></p> <ul style="list-style-type: none"> <li>● Pair visual prompts with verbal presentations</li> <li>● Ask students to restate information, directions, and assignments.</li> <li>● Provide repetition and and practice</li> <li>● Model skills / techniques to be mastered.</li> <li>● Provide extended time to complete class work</li> <li>● Provide copy of class notes</li> <li>● Break long assignments into smaller parts</li> <li>● Assist student in setting short term goals</li> <li>● Allow for preferential seating to be mutually determined by the student and teacher</li> <li>● Provide extra textbooks for home.</li> <li>● Model and reinforce organizational systems (i.e. color-coding)</li> </ul> |   |

- Write out homework assignments, check student's recording of assignments

### Interdisciplinary Connections

#### ELA Standards

- W.9-10.9. Draw evidence from literary or nonfiction informational texts to support analysis, reflection, and research.
- SL.9-10.1. Initiate and participate effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with peers on grades 9–10 topics, texts, and issues, building on others' ideas and expressing their own clearly and persuasively.
- SL.9-10.2. Integrate multiple sources of information presented in diverse media or formats (e.g., visually, quantitatively, qualitatively, orally) evaluating the credibility and accuracy of each source.
- W.11-12.9. Draw evidence from literary or informational texts to support analysis, reflection, and research.
- SL.11-12.1. Initiate and participate effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with peers on grades 11–12 topics, texts, and issues, building on others' ideas and expressing their own clearly and persuasively.
- SL.11-12.2. Integrate multiple sources of information presented in diverse media or formats (e.g., visually, quantitatively, qualitatively, orally) evaluating the credibility and accuracy of each source.

### Career Readiness, Life Literacies and Key Skills

#### Career Readiness, Life Literacies and Key Skills Practices

- Demonstrate creativity and innovation.
- Utilize critical thinking to make sense of problems and persevere in solving them.
- Model integrity, ethical leadership and effective management.
- Plan education and career paths aligned to personal goals.
- Use technology to enhance productivity, increase collaboration and communicate effectively.
- Work productively in teams while using cultural/global competence.

#### Career Readiness, Life Literacies and Key Skills Standards

##### 9.4 Life Literacies and Key Skills

##### 21st Century Skills

- 9.4.12.CI.1. Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2. Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3. Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1. Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
- 9.4.12.CT.2. Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
- 9.4.12.IML.5. Evaluate, synthesize, and apply information on climate change from various sources appropriately (e.g., 2.1.12.CHSS.6, S.IC.B.4, S.IC.B.6, 8.1.12.DA.1, 6.1.12.GeoHE.14.a, 7.1.AL.PRSNT.2).
- 9.4.12.IML.7. Develop an argument to support a claim regarding a current workplace or societal/ethical issue such as climate change.
- 9.4.12.GCA.1. Collaborate with individuals to analyze a variety of potential solutions to climate change effects and determine why solutions may work better than others (e.g., political, economic, cultural).

##### Technology Integration

- 9.4.12.IML.4. Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).

| Climate Change   |
|--|
| <ul style="list-style-type: none"> <li>8.2.12.ETW.3: Identify a complex, global environmental or climate change issue, develop a systemic plan of investigation and propose an innovative sustainable solution.</li> </ul> |
| SEL Competencies   |
| <ul style="list-style-type: none"> <li>Self - Awareness</li> <li>Self - Management</li> <li>Social Awareness</li> <li>Responsible Decision Making</li> <li>Relationship Skills</li> </ul>                                  |
| <p><a href="https://www.nj.gov/education/safety/wellness/selearning/index.shtml">https://www.nj.gov/education/safety/wellness/selearning/index.shtml</a></p>   |

| District/School Formative Assessment Plan   | District/School Summative Assessment Plan  |
|---|--|
| <p><i>Formative assessment informs instruction and is ongoing throughout a unit to determine how students are progressing against the standards.</i></p> <p>Teachers are encouraged to incorporate Formative Assessments into all lessons. During instruction, teachers will collect ongoing information on students' mastery of content through a variety of methods:</p> <ul style="list-style-type: none"> <li>Questioning: using Socratic method, probing questions, a hierarchical system in complexity (Bloom's Taxonomy)</li> <li>Exit tickets, rotational activities (stations), quizzes, and small group activities</li> <li>Classwork, homework, group work (formative assessment)</li> <li>Pre-Assessment, teacher's observation, class discussion, and journal</li> </ul> | <p><i>Summative assessment is an opportunity for students to demonstrate mastery of the skills taught during a particular unit.</i></p> <p><b>Summative Assessment:</b></p> <ul style="list-style-type: none"> <li><b>Divisible by Three Program</b></li> <li><b>Algorithm Tracing Project</b></li> <li><b>EarSketch Ringtone Project</b></li> <li><b>Song Project:</b> Use EarSketch to design a short song that includes loops, effects, tempo, and musical repetition.</li> <li><b>Debugging and Documentation Task:</b> Students debug a sample EarSketch program, then document the changes made.</li> <li><b>Copyright &amp; Ethics Presentation:</b> Students create a short presentation on copyright issues in music and programming.</li> <li><b>Musical Form &amp; Custom Functions Assignment:</b> Write a program in EarSketch that uses at least one custom function to structure a musical piece.</li> </ul> <p><b>Other Summative Assessments:</b> Teachers are encouraged to design and their own assessments (topic/module tests and quizzes) individually and/or with their department or grade-level partners, as per Uniform Grading Profile.</p> |

| Targeted Academic Vocabulary   |
|--|
| <p>loop, repetition, count variable, iteration, range function, for loop, summing, tracing, algorithm, simulation, debugging, documenting, function, parameter, variable, condition, effect, envelope, tempo, pitch, string operations, musical form, beat, repetition, copyright, correctness, EarSketch, creative coding</p> |

| District/School Tasks   | District/School Primary and Supplementary Resources   |
|---|---|
| <ul style="list-style-type: none"> <li>See Formative/Summative Assessments</li> </ul> | <p><b>District-Mandated Resources</b></p> <ul style="list-style-type: none"> <li>Project STEM CS Python Fundamentals</li> </ul> <p><b>Other Resources:</b></p> <ul style="list-style-type: none"> <li><a href="#">Teacher Handbook</a></li> <li><a href="#">Creative Task Examples</a></li> <li><a href="#">Resources from Amazon Future Engineer</a></li> <li><a href="#">The First Programming Languages (Diversity, Equity and Inclusion)</a></li> <li><a href="#">Learn Python</a></li> <li><a href="#">When Women Stopped Coding-CommonLit (Diversity, Equity and Inclusion)</a></li> <li><a href="#">Programming Basics: Statements &amp; Functions (Diversity, Equity and Inclusion)</a></li> <li><a href="#">How Computer Science Can Help Save the Environment (Climate Change)</a></li> <li><a href="#">What Does Computer Science Have To Do With Climate Change (Climate Change)</a></li> </ul> |
| <b>Instructional Best Practices and Exemplars</b>                                     |   |
| See Appendix B  |   |
| Pacing Guide  |   |
| Computer Science I Pacing Guide   |   |

**Unit 3: Programming Structures – Graphics, Functions, and Lists**

**Overview**

In this unit, students will first explore graphics programming, using coordinates, shapes, colors, and animation to visualize code in action. They will then study functions, learning how to define, call, and trace reusable blocks of code with parameters and return values. Finally, students will work with lists, discovering how to declare, access, and manipulate ordered collections of data. Students will apply these concepts in cumulative projects such as designing animations, creating a calendar, and programming a birthday organizer, which demonstrate how graphics, functions, and lists work together to solve real-world problems.

**Essential Questions**

- How can graphics be created and controlled through programming?
- Why are functions important for organizing and simplifying code?
- How do lists help programmers work with large amounts of data?
- In what ways do sorting, searching, and looping make programs more powerful?
- How can graphics, functions, and lists be combined to design efficient and creative solutions?

**Enduring Understandings**

- Programming can generate visual representations through shapes, colors, and animations.
- Functions make programs more efficient and support code reusability.
- Lists are essential for storing and manipulating collections of related data.
- Sorting and searching algorithms provide systematic approaches to organizing information.
- Combining programming structures enables students to create both functional and creative applications.

**Unit 3: Programming Structures – Graphics, Functions, and Lists**

**Content Standards**

- 8.1.12.AP.1. Design algorithms to solve computational problems using a combination of original and existing algorithms.
- 8.1.12.AP.2. Create generalized computational solutions using collections instead of repeatedly using simple variables.

**Unit 3 Mouse Motion Events, Conditionals, and Helper Functions**

**Core Ideas**

- Graphics programming demonstrates how code can be used to create visual representations.
- Functions make programs modular, efficient, and easier to debug by breaking problems into reusable parts.
- Lists allow programmers to store, access, and manipulate collections of data effectively.
- Loops, functions, and lists can be combined to perform complex tasks in fewer lines of code.
- Algorithms such as sorting and searching are essential for managing and organizing data.
- Programming structures support creativity, logical problem-solving, and real-world applications.

**Student Learning Objectives**

Students will be able to...

- Use coordinates, shapes, and color codes to create and animate graphics.
- Define, call, and trace functions with parameters and return values.
- Apply multiple functions to solve programming problems.
- Declare, access, and modify lists using indices and list methods.
- Use loops to iterate over lists for efficiency in coding.
- Apply lists as parameters and explore their connection to strings.
- Implement and evaluate sorting and searching algorithms.
- Design and present projects (Animation, Calendar, Birthday Organizer) that integrate graphics, functions, and lists.

**Integrated Accommodations and Modifications**

| Special Education Students  | English Language Learners   | At Risk   |
|---|---|---|
| <ul style="list-style-type: none"> <li>● Utilize modifications &amp; accommodations delineated in the student’s IEP</li> <li>● Provide additional manipulatives to support instruction</li> <li>● Allow for alternative strategies to solve algorithms or tasks</li> <li>● Provide the steps needed to complete the task</li> <li>● Model frequently</li> <li>● Provide repetition and practice.</li> <li>● Use visuals to demonstrate/model the processes</li> <li>● Restate, reread, and clarify directions/questions</li> <li>● Ask students to restate information, directions, and assignments.</li> <li>● Provide copy of class notes</li> <li>● Distribute study guide for classroom tests.</li> <li>● Provide preferential seating to be mutually determined by the student and teacher</li> <li>● Provide extra textbooks for home.</li> <li>● Provide regular parent/ school communication</li> </ul> | <p>WIDA Can Do Descriptors<br/> <a href="https://wida.wisc.edu/teach/can-do/descriptors">https://wida.wisc.edu/teach/can-do/descriptors</a></p> <ul style="list-style-type: none"> <li>● Modify Assignments</li> <li>● Use testing and portfolio assessment</li> <li>● Utilize Native Language Translation (peer, online assistive technology, translation device, bilingual dictionary)</li> <li>● Repeat, rephrase, paraphrase key concepts and directions</li> <li>● Allow for extended time for assignment completion as needed</li> <li>● Highlight key vocabulary</li> <li>● Define essential vocabulary in context</li> <li>● Use graphic organizers, visuals, manipulatives and other concrete materials</li> <li>● Use gestures, facial expressions and body language</li> <li>● Read aloud</li> <li>● Build on what students already know and prior experience</li> </ul> | <ul style="list-style-type: none"> <li>● Pair visual prompts with verbal presentations</li> <li>● Ask students to restate information, directions, and assignments.</li> <li>● Provide repetition and and practice</li> <li>● Model skills / techniques to be mastered.</li> <li>● Provide extended time to complete class work</li> <li>● Provide copy of class notes</li> <li>● Provide preferential seating to be mutually determined by the student and teacher</li> <li>● Allow the use of a computer to complete assignments.</li> <li>● Establish expectations for correct spelling on assignments</li> <li>● Provide extra textbooks for home.</li> <li>● Provide Peer Support</li> <li>● Increase one on one time</li> </ul> |

|   |  |                                       |
|---|--|---------------------------------------|
| <ul style="list-style-type: none"> <li>● Allow extended time to complete assignment</li> <li>● Establish procedures for accommodations / modifications for assessments</li> <li>● Allow student to take/complete tests in an alternate setting as needed</li> </ul> <p><u>Appendix A: Special Education Accommodations and Modifications</u></p>  |  |                                       |
| <p align="center"><b>Gifted and Talented Students</b></p>   |  | <p align="center"><b>504 Plan</b></p> |
| <ul style="list-style-type: none"> <li>● Utilize advanced, accelerated, or compacted content</li> <li>● Provide assignments that emphasize higher-level thinking skills.</li> <li>● Allow for individual student interest</li> <li>● Gear assignments to development in areas of affect, creativity, cognition, and research skills</li> <li>● Allow for a variety in types of resources</li> <li>● Provide problem-based assignments with planned scope and sequence</li> <li>● Utilize inquiry-based instruction</li> <li>● Adjust the pace of lessons</li> <li>● Utilize Choice Boards</li> <li>● Provide Problem-Based Learning</li> <li>● Establish flexible Grouping</li> </ul>   | <ul style="list-style-type: none"> <li>● Pair visual prompts with verbal presentations</li> <li>● Ask students to restate information, directions, and assignments.</li> <li>● Provide repetition and practice</li> <li>● Model skills / techniques to be mastered.</li> <li>● Provide extended time to complete class work</li> <li>● Provide copy of class notes</li> <li>● Break long assignments into smaller parts</li> <li>● Assist student in setting short term goals</li> <li>● Allow for preferential seating to be mutually determined by the student and teacher</li> <li>● Provide extra textbooks for home.</li> <li>● Model and reinforce organizational systems (i.e. color-coding)</li> <li>● Write out homework assignments, check student's recording of assignments</li> </ul> |                                       |
| <p align="center"><b>Interdisciplinary Connections</b></p>  |  |                                       |
| <p><b>ELA Standards</b></p> <ul style="list-style-type: none"> <li>● W.9-10.9. Draw evidence from literary or nonfiction informational texts to support analysis, reflection, and research.</li> <li>● SL.9-10.1. Initiate and participate effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with peers on grades 9–10 topics, texts, and issues, building on others’ ideas and expressing their own clearly and persuasively.</li> <li>● SL.9-10.2. Integrate multiple sources of information presented in diverse media or formats (e.g., visually, quantitatively, qualitatively, orally) evaluating the credibility and accuracy of each source.</li> <li>● W.11-12.9. Draw evidence from literary or informational texts to support analysis, reflection, and research.</li> <li>● SL.11-12.1. Initiate and participate effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with peers on grades 11–12 topics, texts, and issues, building on others’ ideas and expressing their own clearly and persuasively.</li> <li>● SL.11-12.2. Integrate multiple sources of information presented in diverse media or formats (e.g., visually, quantitatively, qualitatively, orally) evaluating the credibility and accuracy of each source.</li> </ul> |  |                                       |

**Career Readiness, Life Literacies and Key Skills**

**Career Readiness, Life Literacies and Key Skills Practices**

- Demonstrate creativity and innovation.
- Utilize critical thinking to make sense of problems and persevere in solving them.
- Model integrity, ethical leadership and effective management.
- Plan education and career paths aligned to personal goals.
- Use technology to enhance productivity, increase collaboration and communicate effectively.
- Work productively in teams while using cultural/global competence.

**Career Readiness, Life Literacies and Key Skills Standards**

**9.4 Life Literacies and Key Skills**

21st Century Skills

- 9.4.12.CI.1. Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2. Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3. Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1. Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
- 9.4.12.CT.2. Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).

Technology Integration

- 9.4.12.IML.4. Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).

**Climate Change**

**Addressed in Unit 2**

**SEL Competencies**

- **Self - Awareness**
- **Self - Management**
- **Social Awareness**
- **Responsible Decision Making**
- **Relationship Skills**

<https://www.nj.gov/education/safety/wellness/selearning/index.shtml>

| District/School Formative Assessment Plan  | District/School Summative Assessment Plan   |
|--|---|
| <p><i>Formative assessment informs instruction and is ongoing throughout a unit to determine how students are progressing against the standards.</i></p> <p>Teachers are encouraged to incorporate Formative Assessments into all lessons. During instruction, teachers will collect ongoing information on students' mastery of content through a variety of methods:</p> | <p><i>Summative assessment is an opportunity for students to demonstrate mastery of the skills taught during a particular unit.</i></p> <p><b>Summative Assessment:</b></p> |

- Questioning: using Socratic method, probing questions, a hierarchical system in complexity (Bloom’s Taxonomy)
- Exit tickets, rotational activities (stations), quizzes, and small group activities
- Classwork, homework, group work (formative assessment)
- Pre-Assessment, teacher’s observation, class discussion, and journal

- **Animation Project:** Students design and present an animated scene (using shapes, colors, and loops) that demonstrates their understanding of graphics programming.
- **Function-Based Program:** Create a mini-application (e.g., a calendar generator or calculator) that incorporates multiple functions with parameters and return values.
- **Birthday Organizer Project:** Build a list-based program to store and sort birthdays, demonstrating use of lists, searching, and sorting.
- **Portfolio Reflection:** Written explanation of how graphics, functions, and lists work together to build more complex programs.
- **Divisible by Three Program:** Write a program that uses loops and conditionals to solve the problem of finding numbers divisible by three.
- **EarSketch Ringtone Project:** Use EarSketch to design a ringtone incorporating loops, repetition, effects, and rhythm.
- **Debugging and Documentation Report:** Provide a code sample with intentional errors, then submit both the corrected code and a reflection explaining how errors were identified and fixed.
- **Performance Task:** Present a short “music with code” project, explaining the programming structures used and the musical results achieved.
- **Calendar Application:** Students create a calendar program that uses functions and returns values.
- **Data Visualization Project:** Represent a dataset visually through graphics (shapes, colors, or animations).
- **List Sorting Challenge:** Build a program that demonstrates searching and sorting of list data, with a written explanation of the chosen algorithm.
- **Interactive Demo:** Students demonstrate a project (e.g., Birthday Organizer) to the class, explaining design choices and coding structures.

**Other Summative Assessments:** Teachers are encouraged to design and their own assessments (topic/module tests and quizzes) individually and/or with their department or grade-level partners, as per Uniform Grading Profile.

graphics, color code, loop, coordinates, line, circle, animation, function, parameter, return value, trace, modular code, list, element, index, list method, iteration, string, sorting, searching, algorithm, data structure, birthday organizer, calendar

| District/School Tasks   | District/School Primary and Supplementary Resources  |
|---|--|
| <ul style="list-style-type: none"> <li>See Formative/Summative Assessments</li> </ul> | <p><b>District-Mandated Resources</b></p> <ul style="list-style-type: none"> <li>Project STEM CS Python Fundamentals</li> </ul> <p><b>Other Resources:</b></p> <ul style="list-style-type: none"> <li><a href="#">Teacher Handbook</a></li> <li><a href="#">Creative Task Examples</a></li> <li><a href="#">Resources from Amazon Future Engineer</a></li> <li><a href="#">The First Programming Languages (Diversity, Equity and Inclusion)</a></li> <li><a href="#">Learn Python</a></li> <li><a href="#">When Women Stopped Coding-CommonLit (Diversity, Equity and Inclusion)</a></li> <li><a href="#">Programming Basics: Statements &amp; Functions (Diversity, Equity and Inclusion)</a></li> <li><a href="#">7 Black Pioneers in Computer Science (Amistad)</a></li> </ul> |
| <b>Instructional Best Practices and Exemplars</b>                                     |  |
| See Appendix B  |  |
| <b>Pacing Guide</b>   |  |
| Computer Science I Pacing Guide   |  |

**Unit 4: Code in Action – Music, Web, and Careers in Computer Science**

**Overview**

This unit builds on students’ foundational coding knowledge by applying it to new contexts that highlight the versatility of computer science. Students begin with 2D lists, learning how to store, manipulate, and visualize complex data through algorithms, graphics, and animations. Next, they dive into EarSketch, using programming concepts like randomness, data structures, and conditionals to compose music. Students then study the Internet, including IP addressing, packets, routers, and web design with HTML, while also addressing issues of cybersecurity and net neutrality. Finally, students will explore career applications of computer science, connecting their coding knowledge to fields such as data science, medicine, game development, entertainment, social justice, and entrepreneurship. Through hands-on assignments like creating matrices, a song of the summer, and building a webpage, students will see how coding is both a technical and creative discipline with wide-reaching impact.

**Essential Questions**

- How do 2D lists help programmers organize and analyze complex data?
- In what ways can programming be used to create and manipulate music?
- How does the internet work, and why are cybersecurity and net neutrality important?
- How can coding be used to design and publish webpages?
- What career opportunities exist in computer science, and how does coding connect to different fields of work and society?

**Enduring Understandings**

- Programming structures like 2D lists and loops make it possible to model and process large or complex sets of data.
- Coding can be a medium of creative expression, such as composing music with EarSketch.
- The internet relies on protocols, addressing systems, and security measures to function effectively.
- HTML provides a foundation for building and publishing content on the web.
- Ethical considerations such as cybersecurity and net neutrality affect how technology impacts society.
- Computer science skills open pathways into diverse careers that blend technology with human creativity and problem-solving.

**Unit 4: Code in Action – Music, Web, and Careers in Computer Science**

**Content Standards**

- 8.1.12.AP.1. Design algorithms to solve computational problems using a combination of original and existing algorithms.
- 8.1.12.AP.2. Create generalized computational solutions using collections instead of repeatedly using simple variables.

**Unit 4: Code in Action – Music, Web, and Careers in Computer Science**

**Core Ideas**

- Graphics programming demonstrates how code can be used to create visual representations.
- Functions make programs modular, efficient, and easier to debug by breaking problems into reusable parts.
- Lists allow programmers to store, access, and manipulate collections of data effectively.
- Loops, functions, and lists can be combined to perform complex tasks in fewer lines of code.

**Unit 4: Code in Action – Music, Web, and Careers in Computer Science**

**Core Ideas**

- Algorithms such as sorting and searching are essential for managing and organizing data. Programming structures support creativity, logical problem-solving, and real-world applications.

**Student Learning Objectives**

Students will be able to...

- Use coordinates, shapes, and color codes to create and animate graphics.
- Define, call, and trace functions with parameters and return values.
- Apply multiple functions to solve programming problems.
- Declare, access, and modify lists using indices and list methods.
- Use loops to iterate over lists for efficiency in coding.
- Apply lists as parameters and explore their connection to strings.
- Implement and evaluate sorting and searching algorithms.
- Design and present projects (Animation, Calendar, Birthday Organizer) that integrate graphics, functions, and lists.

**Integrated Accommodations and Modifications**

| Special Education Students  | English Language Learners   | At Risk   |
|---|---|---|
| <ul style="list-style-type: none"> <li>● Utilize modifications &amp; accommodations delineated in the student’s IEP</li> <li>● Provide additional manipulatives to support instruction</li> <li>● Allow for alternative strategies to solve algorithms or tasks</li> <li>● Provide the steps needed to complete the task</li> <li>● Model frequently</li> <li>● Provide repetition and practice.</li> <li>● Use visuals to demonstrate/model the processes</li> <li>● Restate, reread, and clarify directions/questions</li> <li>● Ask students to restate information, directions, and assignments.</li> <li>● Provide copy of class notes</li> <li>● Distribute study guide for classroom tests.</li> </ul> | <p>WIDA Can Do Descriptors<br/> <a href="https://wida.wisc.edu/teach/can-do/descriptors">https://wida.wisc.edu/teach/can-do/descriptors</a></p> <ul style="list-style-type: none"> <li>● Modify Assignments</li> <li>● Use testing and portfolio assessment</li> <li>● Utilize Native Language Translation (peer, online assistive technology, translation device, bilingual dictionary)</li> <li>● Repeat, rephrase, paraphrase key concepts and directions</li> <li>● Allow for extended time for assignment completion as needed</li> <li>● Highlight key vocabulary</li> <li>● Define essential vocabulary in context</li> <li>● Use graphic organizers, visuals, manipulatives and other concrete materials</li> <li>● Use gestures, facial expressions and body language</li> <li>● Read aloud</li> </ul> | <ul style="list-style-type: none"> <li>● Pair visual prompts with verbal presentations</li> <li>● Ask students to restate information, directions, and assignments.</li> <li>● Provide repetition and and practice</li> <li>● Model skills / techniques to be mastered.</li> <li>● Provide extended time to complete class work</li> <li>● Provide copy of class notes</li> <li>● Provide preferential seating to be mutually determined by the student and teacher</li> <li>● Allow the use of a computer to complete assignments.</li> <li>● Establish expectations for correct spelling on assignments</li> <li>● Provide extra textbooks for home.</li> <li>● Provide Peer Support</li> <li>● Increase one on one time</li> </ul> |

|  |  |  |
|--|--|--|
| <ul style="list-style-type: none"> <li>● Provide preferential seating to be mutually determined by the student and teacher</li> <li>● Provide extra textbooks for home.</li> <li>● Provide regular parent/ school communication</li> <li>● Allow extended time to complete assignment</li> <li>● Establish procedures for accommodations / modifications for assessments</li> <li>● Allow student to take/complete tests in an alternate setting as needed</li> </ul> <p><u>Appendix A: Special Education Accommodations and Modifications</u></p> | <ul style="list-style-type: none"> <li>● Build on what students already know and prior experience</li> </ul> |  |
|--|--|--|

| <b>Gifted and Talented Students</b>  | <b>504 Plan</b>  |
|--|--|
| <ul style="list-style-type: none"> <li>● Utilize advanced, accelerated, or compacted content</li> <li>● Provide assignments that emphasize higher- level thinking skills.</li> <li>● Allow for individual student interest</li> <li>● Gear assignments to development in areas of affect, creativity, cognition, and research skills</li> <li>● Allow for a variety in types of resources</li> <li>● Provide problem-based assignments with planned scope and sequence</li> <li>● Utilize inquiry-based instruction</li> <li>● Adjust the pace of lessons</li> <li>● Utilize Choice Boards</li> <li>● Provide Problem-Based Learning</li> <li>● Establish flexible Grouping</li> </ul> | <ul style="list-style-type: none"> <li>● Pair visual prompts with verbal presentations</li> <li>● Ask students to restate information, directions, and assignments.</li> <li>● Provide repetition and and practice</li> <li>● Model skills / techniques to be mastered.</li> <li>● Provide extended time to complete class work</li> <li>● Provide copy of class notes</li> <li>● Break long assignments into smaller parts</li> <li>● Assist student in setting short term goals</li> <li>● Allow for preferential seating to be mutually determined by the student and teacher</li> <li>● Provide extra textbooks for home.</li> <li>● Model and reinforce organizational systems (i.e. color-coding)</li> <li>● Write out homework assignments, check student's recording of assignments</li> </ul> |

**Interdisciplinary Connections**

|  |
|--|
| <p><b>ELA Standards</b></p> <ul style="list-style-type: none"> <li>● W.9-10.9. Draw evidence from literary or nonfiction informational texts to support analysis, reflection, and research.</li> <li>● SL.9-10.1. Initiate and participate effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with peers on grades 9–10 topics, texts, and issues, building on others’ ideas and expressing their own clearly and persuasively.</li> <li>● SL.9-10.2. Integrate multiple sources of information presented in diverse media or formats (e.g., visually, quantitatively, qualitatively, orally) evaluating the credibility and accuracy of each source.</li> <li>● W.11-12.9. Draw evidence from literary or informational texts to support analysis, reflection, and research.</li> </ul> |
|--|

- SL.11-12.1. Initiate and participate effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with peers on grades 11–12 topics, texts, and issues, building on others’ ideas and expressing their own clearly and persuasively.
- SL.11-12.2. Integrate multiple sources of information presented in diverse media or formats (e.g., visually, quantitatively, qualitatively, orally) evaluating the credibility and accuracy of each source.

**Career Readiness, Life Literacies and Key Skills**

**Career Readiness, Life Literacies and Key Skills Practices**

- Demonstrate creativity and innovation.
- Utilize critical thinking to make sense of problems and persevere in solving them.
- Model integrity, ethical leadership and effective management.
- Plan education and career paths aligned to personal goals.
- Use technology to enhance productivity, increase collaboration and communicate effectively.
- Work productively in teams while using cultural/global competence.

**Career Readiness, Life Literacies and Key Skills Standards**

**9.4 Life Literacies and Key Skills**

21st Century Skills

- 9.4.12.CI.1. Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
- 9.4.12.CI.2. Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
- 9.4.12.CI.3. Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
- 9.4.12.CT.1. Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
- 9.4.12.CT.2. Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).

Technology Integration

- 9.4.12.IML.4. Assess and critique the appropriateness and impact of existing data visualizations for an intended audience (e.g., S-ID.B.6b, HS-LS2-4).

**Climate Change**

**Addressed in Unit 2**

**SEL Competencies**

- **Self - Awareness**
- **Self - Management**
- **Social Awareness**
- **Responsible Decision Making**
- **Relationship Skills**

<https://www.nj.gov/education/safety/wellness/selearning/index.shtml>

*Formative assessment informs instruction and is ongoing throughout a unit to determine how students are progressing against the standards.*

Teachers are encouraged to incorporate Formative Assessments into all lessons. During instruction, teachers will collect ongoing information on students' mastery of content through a variety of methods:

- Questioning: using Socratic method, probing questions, a hierarchical system in complexity (Bloom's Taxonomy)
- Exit tickets, rotational activities (stations), quizzes, and small group activities
- Classwork, homework, group work (formative assessment)
- Pre-Assessment, teacher's observation, class discussion, and journal

*Summative assessment is an opportunity for students to demonstrate mastery of the skills taught during a particular unit.*

**Summative Assessment:**

- **Animation Project:** Students design and present an animated scene (using shapes, colors, and loops) that demonstrates their understanding of graphics programming.
- **Function-Based Program:** Create a mini-application (e.g., a calendar generator or calculator) that incorporates multiple functions with parameters and return values.
- **Birthday Organizer Project:** Build a list-based program to store and sort birthdays, demonstrating use of lists, searching, and sorting.
- **Portfolio Reflection:** Written explanation of how graphics, functions, and lists work together to build more complex programs.
- **Divisible by Three Program:** Write a program that uses loops and conditionals to solve the problem of finding numbers divisible by three.
- **EarSketch Ringtone Project:** Use EarSketch to design a ringtone incorporating loops, repetition, effects, and rhythm.
- **Debugging and Documentation Report:** Provide a code sample with intentional errors, then submit both the corrected code and a reflection explaining how errors were identified and fixed.
- **Performance Task:** Present a short "music with code" project, explaining the programming structures used and the musical results achieved.
- **Calendar Application:** Students create a calendar program that uses functions and returns values.
- **Data Visualization Project:** Represent a dataset visually through graphics (shapes, colors, or animations).
- **List Sorting Challenge:** Build a program that demonstrates searching and sorting of list data, with a written explanation of the chosen algorithm.
- **Interactive Demo:** Students demonstrate a project (e.g., Birthday Organizer) to the class, explaining design choices and coding structures.
- **Matrices Project:** Apply 2D lists to solve a problem, such as animating graphics or modeling data.
- **Song of the Summer Project (EarSketch):** Compose a multi-layered music track using loops, randomness, data structures, and effects.

|  |  |
|--|--|
|  | <ul style="list-style-type: none"> <li>● <b>Build Your Own Webpage:</b> Create and publish a basic webpage using HTML with multiple elements (headings, links, images, lists, etc.).</li> <li>● <b>Cybersecurity Case Study:</b> Analyze a real-world cybersecurity scenario and present strategies for protecting data.</li> <li>● <b>Career Research Presentation:</b> Students choose one computer science career and present how programming applies to that field, connecting it to skills learned in the unit.</li> </ul> <p><b>Other Summative Assessments:</b> Teachers are encouraged to design and their own assessments (topic/module tests and quizzes) individually and/or with their department or grade-level partners, as per Uniform Grading Profile.</p> |
|--|--|

**Targeted Academic Vocabulary**

graphics, color code, loop, coordinates, line, circle, animation, function, parameter, return value, trace, modular code, list, element, index, list method, iteration, string, sorting, searching, algorithm, data structure, birthday organizer, calendar

| District/School Tasks   | District/School Primary and Supplementary Resources  |
|---|--|
| <ul style="list-style-type: none"> <li>● See Formative/Summative Assessments</li> </ul> | <p><b>District-Mandated Resources</b></p> <ul style="list-style-type: none"> <li>● Project STEM CS Python Fundamentals</li> </ul> <p><b>Other Resources:</b></p> <ul style="list-style-type: none"> <li>● <a href="#">Teacher Handbook</a></li> <li>● <a href="#">Creative Task Examples</a></li> <li>● <a href="#">Resources from Amazon Future Engineer</a></li> <li>● <a href="#">The First Programming Languages (Diversity, Equity and Inclusion)</a></li> <li>● <a href="#">Learn Python</a></li> <li>● <a href="#">When Women Stopped Coding-CommonLit (Diversity, Equity and Inclusion)</a></li> <li>● <a href="#">Programming Basics: Statements &amp; Functions (Diversity, Equity and Inclusion)</a></li> </ul> |

**Instructional Best Practices and Exemplars**

See Appendix B

**Pacing Guide**

**Computer Science I Pacing Guide**

## **Appendix A: Accommodations and Modifications**

**Appendix A: Accommodations and Modifications: Unit 1**

**Appendix A: Accommodations and Modifications: Unit 2**

**Appendix A: Accommodations and Modifications: Unit 3**

**Appendix A: Accommodations and Modifications: Unit 4**

## **Appendix B: Exemplars and Explanations**

**Appendix B: Instructional Exemplars and Explanations: Unit 1**

**Appendix B: Instructional Exemplars and Explanations: Unit 2**

**Appendix B: Instructional Exemplars and Explanations: Unit 3**

**Appendix B: Instructional Exemplars and Explanations: Unit 4**

## **Appendix C: Classroom Philosophy, Schedule, Structure, and Expectations**