

## CompuScholar, Inc.

### Alignment to the Tennessee K-12 Computer Science Standards

#### 6th - 8th Grade (Middle School)

#### Tennessee Standards:

<b>Name:</b>	K-12 Computer Science Standards
<b>Grade Level:</b>	6 - 8
<b>Standards Link:</b>	<a href="#">K-12 Computer Science Standards (October 2022)</a>

#### CompuScholar Courses:

<b>Course Title:</b>	Tech Essentials	Python Programming
<b>Course ISBN:</b>	978-1-946113-03-0	978-1-946113-00-9
<b>Course Year:</b>	2025	2025

### Description

The Tennessee middle school Computer Science standards are organized into 6 major core concepts. Elements from our **Tech Essentials** and **Python Programming** courses can be used to meet these requirements. Both courses support overlapping skills, and teachers can select the best chapters for their classrooms.

### Syllabus and Pacing Guide to Meet State Requirements

To meet "K-12 Computer Science" requirements for 6th - 8th grades (middle school) over a full school year, schools can use the entire "**Tech Essentials**" course for the first semester and the entire "**Python Programming**" course in the second semester.

To complete requirements in a more **accelerated timeframe**, CompuScholar suggests using the following chapters from each course. Alternate sequences may be used at teacher discretion.

<b>Tech Essentials</b>	<b>Python Programming</b>
Chapter 1	Chapter 1
Chapter 2	Chapter 2
Chapter 5	Chapter 3
Chapter 8	Chapter 4
Chapter 9	Chapter 5
Chapter 10	Chapter 6
Chapter 11	Chapter 7
Chapter 12	Chapter 8
Chapter 13	Chapter 9
Suppl. Chapter 3	Chapter 10 (advanced students)
	Chapter 11 (advanced students)
	Chapter 12 (advanced students)
	Chapter 13 (advanced students)

## Computer Science Standards

**Note 1:** Citation(s) listed may represent a subset of the instances where objectives are met throughout the course.

**Note 2:** Citation(s) for a "Lesson" refer to the "Lesson Text" elements and associated "Activities" within the course, unless otherwise noted. The "Instructional Video" components are supplements designed to introduce or re-enforce the main lesson concepts, and the Lesson Text contains full details.

**Note 3:** Citations to "Supplemental" or "Suppl." refer to Supplemental chapters found at the ends of each course.

<b>MS.FC: Foundational Concepts</b>	<b>TECH ESSENTIALS CITATION(S)</b>	<b>PYTHON PROGRAMMING CITATION(S)</b>
1) Analyze the advantages and limitations of existing computing devices to improve user experience.	Chapter 1, Lesson 4 Suppl. Chapter 1, Lesson 1	
2) Demonstrate skills in identifying and solving hardware and software problems that can occur during regular usage.	Chapter 5, Lesson 2 Chapter 10, Lesson 4	Chapter 5
3) Apply computational thinking to a variety of problems across multiple disciplines.	Chapter 2, Lessons 3, 4 Chapters 8, 9, 10	Chapter 13 Suppl. Ch. 3, Lesson 3 Suppl. Ch. 4, Lesson 4
4) Understand how collaboration is essential to computer science and apply collaborative skills to develop computational solutions.	Chapters 7, 11	Chapter 13

<b>MS.AT: Algorithmic Thinking</b>	<b>TECH ESSENTIALS CITATION(S)</b>	<b>PYTHON PROGRAMMING CITATION(S)</b>
1) Use clearly named variables of various data types to create generalized algorithms.	Chapter 8, Lessons 3, 4	Chapter 2
2) Create algorithms which include methods of controlling the flow of computation using "if...then... else" type conditional statements to perform different operations depending on the values of inputs.	Chapter 9, Lesson 2	Chapter 4
3) Identify algorithms that make use of sequencing, selection, or iteration.	Chapter 9, Lessons 3, 4, 5	Chapters 3, 4, 6
4) Describe how algorithmic processes and automation increase efficiency.	Chapter 10, Lesson 1	Suppl. Ch. 3, Lesson 2 Suppl. Ch. 4, Lesson 4

<b>MS.DA: Data Analysis</b>	<b>TECH ESSENTIALS CITATION(S)</b>	<b>PYTHON PROGRAMMING CITATION(S)</b>
1) Represent data using multiple encoding schemes, such as decimal, binary, Unicode, Morse code, Shorthand, student-created codes.	Chapter 8, Lesson 2 (decimal/binary) Suppl. Ch. 2, Lesson 2 (decimal/binary)	Suppl. Ch. 3, Lessons 1, 2

2) Refine computational models based on the data they have generated.	Suppl. Ch. 3, Lesson 2	
3) Collect, analyze, transform, and refine computational data to make it more useful and reliable.	Suppl. Ch. 3, Lessons 2, 3, 4, 5	

<b>MS.NI: Networking and the Internet</b>	<b>TECH ESSENTIALS CITATION(S)</b>	<b>PYTHON PROGRAMMING CITATION(S)</b>
1) Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.	Chapter 5, Lesson 2 Chapter 10, Lesson 4	Chapter 5
2) Differentiate between secure and non-secure websites and applications including how they affect and use personal data.	Chapter 12, Lesson 1 Chapter 13, Lessons 1, 2	Suppl. Ch. 2, Lessons 3, 4
3) Describe the causes and effects of intellectual property as it relates to print and digital media, considering copyright, fair use, licensing, sharing, and attribution.	Chapter 12, Lesson 3	Suppl. Ch. 2, Lesson 2
4) Compare and contrast common methods of securing data and cybersecurity.	Chapter 13, Lessons 1, 2	Suppl. Ch. 2, Lesson 3
5) Analyze different modes of social engineering and their effectiveness.	Chapter 13, Lesson 1	Suppl. Ch. 2, Lesson 4

<b>MS.PC: Programming Concepts</b>	<b>TECH ESSENTIALS CITATION(S)</b>	<b>PYTHON PROGRAMMING CITATION(S)</b>
1) Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	Chapter 10, Lessons 2, 3	Chapters 9, 10, 11
2) Create procedures with parameters that hide the complexity of a task and can be reused to solve similar		Chapter 9
3) Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	Chapter 11, Lesson 3 / Activity 4	Chapter 13, Activity 4
4) Provide proper attribution when incorporating existing code, media, and libraries into original programs.		Chapter 7 (importing libraries)
5) Use the iterative design process to systematically test and refine programs to improve performance and eliminate errors.	Chapter 11	Chapter 13, Activity 4
6) Document programs using comments and/or README files to make them easier to follow, test, and debug.		Chapter 1, Lesson 3
7) Design a function using a programming language.		Chapter 9