CompuScholar, Inc.

Alignment to the Tennessee **K-12 Computer Science Standards**
6th - 8th Grade (Middle School)

Tennessee Standards:

Name:	K-12 Computer Science Standards
Grade Level:	6 - 8
Standards Link:	K-12 Computer Science Standards (October 2022)

CompuScholar Courses:

Course Title:	Digital Savvy	Python Programming
Course ISBN:	978-0-9887070-8-5	978-1-946113-00-9
Course Year:	2024	2024

Description

The Tennessee middle school Computer Science standards are organized into 6 major core concepts. Elements from our **Digital Savvy** 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), CompuScholar suggests using the following chapters of our "**Digital Savvy**" and "**Python Programming**" courses. Alternate sequences may be used at teacher discretion.

Digital Savvy	**Python Programming**
Chapter 1	Chapter 1
Chapter 2	Chapter 2
Chapter 5	Chapter 3
Chapter 6	Chapter 4
Chapter 8	Chapter 5
Chapter 10	Chapter 6
Chapter 12	Chapter 7
Chapter 13	Chapter 8
Chapter 14	Chapter 9
Chapter 17	Chapter 10 (advanced students)
Chapter 18	Chapter 11 (advanced students)
Chapter 22	Chapter 12 (advanced students)
Chapter 23	Chapter 13 (advanced students)
Chapter 25	
Suppl. Chapter 1	
Suppl. Chapter 2	
Suppl. Chapter 3	

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.

MS.FC: Foundational Concepts	**DIGITAL SAVVY CITATION(S)**	**PYTHON PROGRAMMING CITATION(S)**
1) Analyze the advantages and limitations of existing computing devices to improve user experience.	Chapter 1, Lesson 2	
2) Demonstrate skills in identifying and solving hardware and software problems that can occur during regular usage.	Chapter 5, Lesson 3 Chapter 6, Lesson 4	Chapter 5
3) Apply computational thinking to a variety of problems across multiple disciplines.	Chapter 10, Lessons 5, 6, 7 Chapters 22, 23	Chapter 13 Suppl. Chapter 3, Lesson 3 Suppl. Chapter 4, Lesson 4
4) Understand how collaboration is essential to computer science and apply collaborative skills to develop computational solutions.	Chapters 13, 14, 25	Chapter 13

MS.AT: Algorithmic Thinking	**DIGITAL SAVVY CITATION(S)**	**PYTHON PROGRAMMING CITATION(S)**
1) Use clearly named variables of various data types to create generalized algorithms.	Chapter 23, Lesson 1	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 23, Lessons 2, 3	Chapter 4
3) Identify algorithms that make use of sequencing, selection, or iteration.	Chapter 22, Lesson 3 Chapter 23, Lessons 2, 3	Chapters 3, 4, 6
4) Describe how algorithmic processes and automation increase efficiency.	Suppl. Chapter 2, Lesson 2	Suppl. Chapter 3, Lesson 2 Suppl. Chapter 4, Lesson 4

MS.DA: Data Analysis	**DIGITAL SAVVY CITATION(S)**	**PYTHON PROGRAMMING CITATION(S)**
1) Represent data using multiple encoding schemes, such as decimal, binary, Unicode, Morse code, Shorthand, student-created codes.	Suppl. Chapter 2, Lessons 1, 3	Suppl. Chapter 3, Lessons 1, 2

2) Refine computational models based on the data they have generated.	Suppl. Chapter 2, Lesson 5	
3) Collect, analyze, transform, and refine computational data to make it more useful and reliable.	Chapter 10, Lessons 5, 6, 7 Chapters 14, 25	

MS.NI: Networking and the Internet	DIGITAL SAVVY CITATION(S)	PYTHON PROGRAMMING CITATION(S)
1) Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.	Chapter 5, Lesson 3 Chapter 6, Lesson 4	Chapter 5
2) Differentiate between secure and non-secure websites and applications including how they affect and use personal data.	Chapter 6, Lesson 6 Chapter 8, Lessons 1, 2 Chapters 17, 18	Suppl. Chapter 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 8, Lesson 5	Suppl. Chapter 2, Lesson 2
4) Compare and contrast common methods of securing data and cybersecurity.	Chapter 8, Lessons 2, 3	Suppl. Chapter 2, Lesson 3
5) Analyze different modes of social engineering and their effectiveness.	Suppl. Chapter 1, Lesson 2	Suppl. Chapter 2, Lesson 4

MS.IC: Impacts of Computing	DIGITAL SAVVY CITATION(S)	PYTHON PROGRAMMING CITATION(S)
1) Identify and evaluate the impacts computer science innovations have had on our society.	Chapter 2, Lesson 5 Suppl. Chapter 1, Lesson 1 Suppl. Chapter 3, Lesson 3	Suppl. Chapter 4, Lessons 1, 3, 4
2) Identify how computational systems are being used to collect and analyze information both public and private and understand the benefits and disadvantages of these systems for the user and developer.	Chapter 8, Lesson 1 Chapters 17, 18	Suppl. Chapter 4, Lesson 2
3) Cite evidence of the positive and negative effects of data permanence on personal and professional digital identity.	Chapter 8, Lesson 1 Chapters 17, 18	Suppl. Chapter 4, Lesson 2
4) Discuss digital globalization and Internet censorship.	Suppl. Chapter 1, Lessons 1, 5	Suppl. Chapter 4, Lessons 1, 3
5) Investigate a variety of education pathways and career options that utilize computational thinking and/or computer science skills across the state of Tennessee and the world.	Chapter 24, Lesson 1 Suppl. Chapter 3, Lesson 2	Suppl. Chapter 3, Lessons 4, 5

MS.PC: Programming Concepts	DIGITAL SAVVY CITATION(S)	PYTHON PROGRAMMING CITATION(S)
1) Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	Chapter 22, Lesson 3 Suppl. Chapter 2, Lesson 1	Chapters 9, 10, 11
2) Create procedures with parameters that hide the complexity of a task and can be reused to solve similar tasks.		Chapter 9
3) Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	Chapter 25, Activity 3	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 25, Activity 3 Suppl. Chapter 2, Lesson 6	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