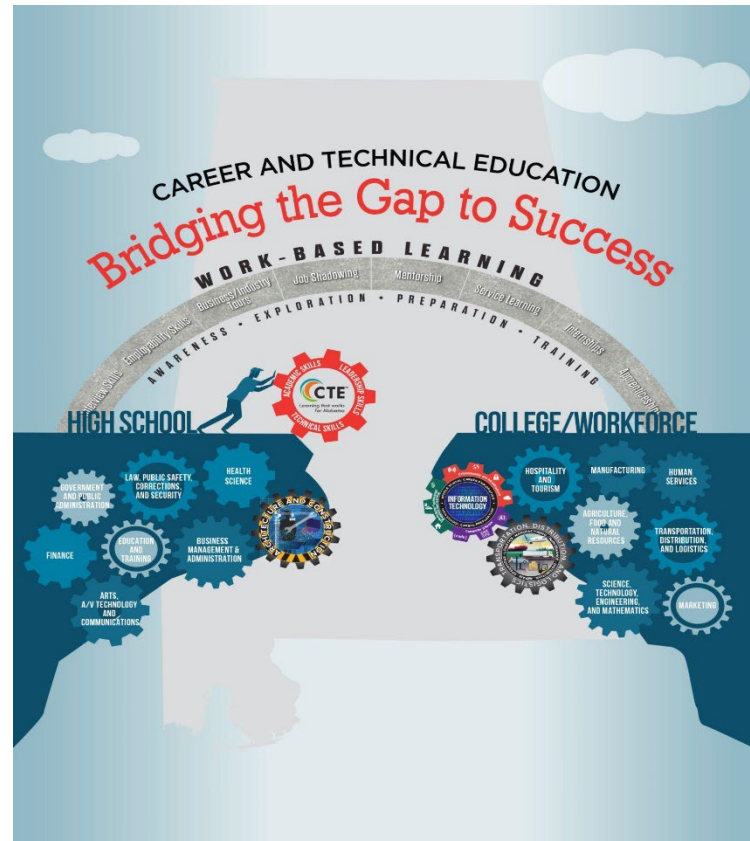


# Information Technology Rubric

Publisher Name: CompuScholar, Inc.



2022

Eric G. Mackey, State Superintendent of  
Education Alabama State Department of  
Education

Publisher Name: CompuScholar, Inc. "Python Programming" Course

**CompuScholar Course Details:**

<b>Course Title:</b>	Python Programming
<b>Course ISBN:</b>	978-1-946113-00-9
<b>Course Year:</b>	2022

**Publisher Note 1:** 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 can be ignored for alignment purposes.

**Publisher Note 2:** Citation(s) to "Supplemental" or "Suppl." chapters refer to Supplemental Chapters found at the end of the main sequence of numbered chapters within the course.

## PROGRAMMING FOUNDATIONS CONTENT STANDARDS

Each content standard completes the stem “*Students will...*”

Topic	Content Standard	Location of Standard in Resource (chapters, sections, pages, etc.)
Digital Literacy	<ol style="list-style-type: none"> <li>1. Describe ethical and legal practices for safeguarding the confidentiality of business-related information.</li> <li>2. Describe possible threats to a laptop, tablet, computer, and/or network and methods for avoiding attacks related to programming.</li> <li>3. Explain the consequences of social engineering, illegal, and unethical uses of technology. <i>Examples: piracy, illegal downloading, licensing infringement; inappropriate use of software, hardware, or mobile devices in the work environment</i></li> <li>4. Describe computing innovations which have the potential to advance programming or other aspects of computer science. <i>Examples: artificial intelligence, quantum computing, low- or no-code programming</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Suppl. Chapter 2, Lessons 1 - 4 Text</li> <li>2. Suppl. Chapter 2, Lessons 3, 4 Text</li> <li>3. Suppl. Chapter 2, Lessons 1, 4 Text</li> <li>4. Suppl. Chapter 4, Lesson 4 Text</li> </ol>
Computer Systems	<ol style="list-style-type: none"> <li>5. Describe the flow of data and instructions through computer systems.</li> <li>6. Explain how data is represented, manipulated, and stored in a computer.</li> </ol>	<ol style="list-style-type: none"> <li>5. Chapter 1, Lesson 3 Text (Order of Execution)</li> <li>6. Suppl. Chapter 3, Lessons 1, 2 Text</li> </ol>

	<p>7. Describe the components of the programming development environment (the hardware and software used by programmers). <i>Examples: text editor, compiler, debugging, profiler, IDE, modeling</i></p>	<p>7. Chapter 1, Lesson 2 Text Chapter 5, Lesson 3 Text Suppl. Chapter 1, Lesson 3 Text</p>
<p><b>Software Design and Programming</b></p>	<p>8. Compare and contrast current programming languages utilized by business and industry and determine features, functions, and benefits of each.</p> <p>9. Identify and explain various kinds of cryptographic algorithms. <i>Examples: hashing, symmetric, asymmetric</i></p> <p>10. Explain why any input-processing algorithm must correctly handle all problem variants.</p> <p>11. Write an algorithm to solve mathematical problems using formulas, equations, and functions.</p> <p>12. Represent the logical flow of a program graphically. <i>Examples: flowcharts, data traces, input/output charts</i></p> <p>13. Utilize and explain techniques for code commenting and documentation. <i>Example: inserting meta text in source code</i></p> <p>14. Design a program that uses mathematical operations, data, functions, looping and iteration, sequencing, abstraction, lists, and selection. <i>Examples: if-else statements, comparison</i></p>	<p>8. N/A</p> <p>9. Suppl. Chapter 3, Lesson 2 Text</p> <p>10. Chapter 8, Lesson 3 Text</p> <p>11. Chapter 10 Activity</p> <p>12. Suppl. Chapter 3, Lesson 3 Text</p> <p>13. Chapter 2, Lesson 3 Text</p> <p>14. Chapter 2, Lesson 2 Text (Math &amp; data) Chapter 9, Lessons 1, 2 Text (Functions) Chapter 6, Lessons 3, 4 Text (Loops) Chapter 1, Lesson 3 Text (Sequential execution) Chapters 10, 11 (OOP / abstraction) Chapter 6, Lessons 1, 2 Text (Lists) Chapter 4, Lesson 2 Text (branching/selection with if()) Chapter 13 (Creative program design)</p>

	<p>a. Design a program using visual modeling software to illustrate abstraction of languages from the solutions.  <i>Examples: Appian, Claris FileMaker, DWkit, Google AppSheet, Looker 7, Mendix, Microsoft PowerApps, OutSystems, Robocoder Rintagi, Salesforce Lightning, Sisense, Skyve Foundry, Temenos (formerly Kony), SIB Visions VisionX, Wix Editor X, Yellowfin 9, Zoho Creator</i></p> <p>15. Design a program that passes arguments and parameters (variables).</p> <p>16. Evaluate algorithms based on given designs to discuss their efficiency, correctness, and clarity.  <i>Examples: analyzing and comparing execution times, testing with multiple inputs or data sets, debugging</i></p> <p>17. Construct programs that utilize logical algorithms from specifications and requirement statements.</p> <p>18. Create a model software program which involves coding, testing, and documenting according to industry coding standards and guidelines.</p>	<p>14.a N/A (Visual Modeling)</p> <p>15. Chapter 9, Lesson 2 Text</p> <p>16. Suppl. Chapter 3, Lesson 3 Text</p> <p>17. Suppl. Chapter 3, Lesson 3 Text            Chapter 12, and all other chapter activities</p> <p>18. Chapter 13, Lesson 1, Activities 1 - 4</p>
--	---	---

<p><b>Computer Numbering Systems</b></p>	<p>18. Explain how strings of 0s and 1s are used in programming.</p> <p>19. Summarize how numerical values are represented using different bases, including decimal and binary.</p> <p>20. Demonstrate how numbers with decimals can have fixed-point or floating-point representations in binary.</p> <p>21. Compare and contrast quantum and classical computing notation systems. <i>Example: qubits</i></p>	<p>15. Suppl. Chapter 3, Lesson 1 Text</p> <p>16. Suppl. Chapter 3, Lessons 1 – 2 Text</p> <p>17. N/A</p> <p>18. N/A</p>
--	---	--