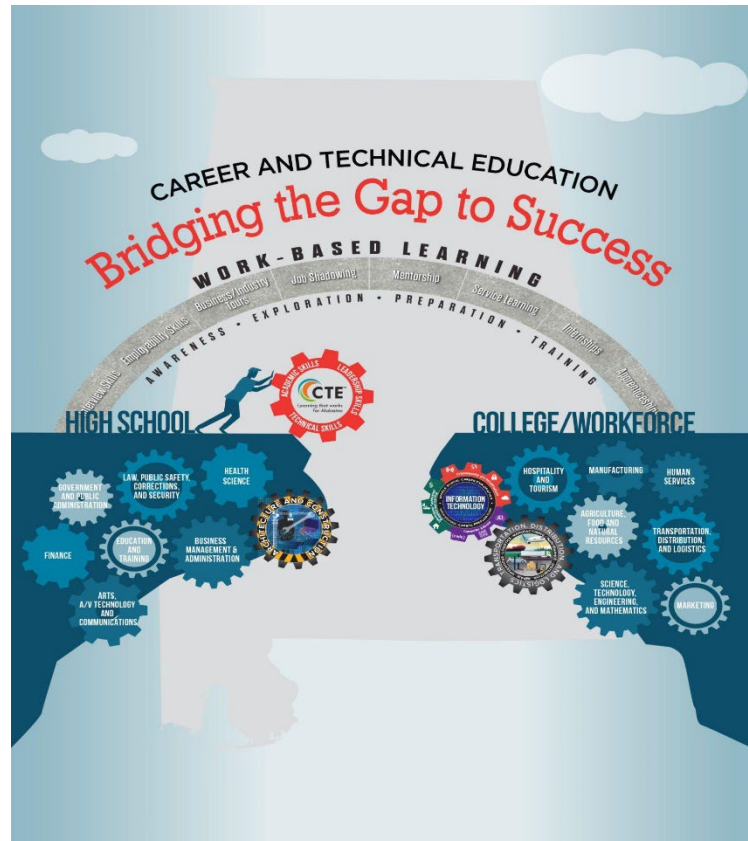


Information Technology Rubric

Publisher Name: CompuScholar, Inc.



2022

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

Publisher Name: CompuScholar, Inc. – “C# Programming” Course

CompuScholar Course Details:

Course Title:	C# Programming
Course ISBN:	978-1-946113-01-6
Course Year:	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.

OBJECT-ORIENTED PROGRAMMING I CONTENT STANDARDS

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

Topic	Content Standard	Location of Standard in Resource (chapters, sections, pages, etc.)
Technical Knowledge	1. Describe the differences between structured programming and object-oriented programming (OOP).	1. Chapter 13, Lesson 1 Text
Object-Oriented Programming Foundations	2. Describe the use of a main method in an application. <i>Examples: signature of main, how to consume an instance of your own class, command line arguments</i> 3. Perform basic input and output using standard input and output streams. <i>Examples: print statements, import and use the scanner class</i> 4. Evaluate the scope of a variable and declare a variable within a block, class, or method.	2. Chapter 2, Lesson 2 Text 3. Chapter 3, Lesson 3 Text Chapter 5, Lesson 2 Text 4. Chapter 14, Lesson 4 Text

**Object-Oriented
Language-Specific
Basics**

- | | |
|---|---|
| <ol style="list-style-type: none"> 5. Define the inputs and outputs of a computer program. 6. Declare and use primitive data type variables.
<i>Examples: byte, char, int, double, short, long, float, Boolean</i> <ol style="list-style-type: none"> a. Identify precision loss in primitive data types.
<i>Examples: double, float</i> b. Explain how primitives differ from wrapper object types.
<i>Examples: integer, Boolean</i> 7. Outline and explain the logical steps necessary for the development of a computer program.
<i>Examples: software development life cycle, structured programming</i> 8. Create a basic algorithm using plain language (pseudocode). 9. Use flowcharts to represent logic graphically. 10. Explain the basic functions of the integrated development environment (IDE).
<i>Examples: inputting the code, saving the code, executing the code</i> | <ol style="list-style-type: none"> 5. Chapter 3, Lesson 3 Text
Chapter 5, Lesson 2 Text 6. Chapter 3, Lessons 1, 2 Text
Chapter 4, Lesson 3 Text 7. Chapter 12, Lessons 1, 2 Text
Chapter 24, Lessons 1, 2 Text 8. Chapter 12, Lesson 2 Text 9. Chapter 12, Lesson 1 Text 10. Chapter 1, Lesson 3 Text
Chapter 2, Lesson 2 Text
Chapter 10, Lesson 2 Text
Chapter 25, Lesson 3 Text |
|---|---|

**Object-Oriented
Language-Specific
Syntax**

- | | |
|--|--|
| <p>11. Describe the functions of different objects and their purposes in a program.</p> <p>12. Describe the function and purpose of a computer program event procedure.</p> <p>13. Write property assignment statements in computer code.</p> <p>14. Utilize comments in the program code to document the logic and enhance the readability of the code.</p> <p>15. List and explain computer program operator types and the precedence (order of operation) of program operators.
<i>Examples: Boolean, relational, arithmetical</i></p> <p>16. Differentiate between commands and statements in computer programming.</p> <p>17. Write valid declaration statements using an approach to global static scope with appropriate data types, including constants, variables, and logical reasoning.</p> | <p>11. Chapter 13, Lessons 1, 2 Text</p> <p>12. Chapter 26, Lesson 2 Text</p> <p>13. Chapter 14, Lessons 1, 3 Text</p> <p>14. Chapter 2, Lesson 3 Text</p> <p>15. Chapter 4, Lesson 1 Text
Chapter 7, Lesson 1 Text
Chapter 8, Lessons 1, 2 Text</p> <p>16. Chapter 2, Lesson 3 Text</p> <p>17. Chapter 3, Lesson 2 Text
Chapter 14, Lessons 1, 4 Text
Chapter 15, Lesson 1 Text</p> |
|--|--|

Practical
Programming

- | | |
|---|---|
| <p>18. Construct a program that will perform calculations on a set of given data.</p> <p>19. Generate random numbers through the use of built-in functions in a program.</p> <p>20. Utilize accumulators and counters in a program.</p> <p>21. Identify and utilize various looping and iteration structures that control the flow of a program.</p> <p>22. Utilize built-in properties and functions to manipulate classes and structures within a program.
<i>Example: string</i></p> <p>23. Describe the purpose and function of a class.</p> <p>24. Construct and evaluate class definitions.
<i>Examples: constructors, constructor overloading, one class per java file, this keyword, basic inheritance, overriding</i></p> <p>25. Describe the purposes and functions of general sub procedures in a program.</p> <p>26. Explain the uses of parameters and arguments and how they control the flow of a program.</p> <p>27. Create a program using one or more classes and/or functions.</p> <p>28. Create a program using a general sub procedure passing arguments to another sub procedure.</p> <p>29. Construct and evaluate code that uses branching statements.
<i>Examples: if, else, else if, switch; single-line vs. block;</i></p> | <p>18. Chapter 4, Lesson 1 Text
Chapter 4 Activity</p> <p>19. Chapter 6, Lesson 2 Text</p> <p>20. Chapter 11, Lesson 2 Text
Chapter 11 Activity
...and most places with iteration</p> <p>21. Chapter 11</p> <p>22. Chapter 5, Lessons 1, 2 Text
Chapter 6, Lesson 2 Text</p> <p>23. Chapter 13, Lesson 1 Text</p> <p>24. Chapters 13, 14, 15, 21, 22</p> <p>25. Chapter 13, Lesson 3 Text</p> <p>26. Chapter 13, Lesson 4 Text</p> <p>27. Chapters 13, 14</p> <p>28. Chapter 13, Lesson 4 Text
Chapter 13 Activity (doTrick() method calls overloaded doTrick() version)</p> <p>29. Chapters 7, 8</p> |
|---|---|

	<p style="text-align: center;"><i>nesting; logical, relational operators</i></p> <p>30. Construct and evaluate code that uses loops. <i>Examples: while, for, for each, do while; break and continue; nesting; logical, relational, and unary operators</i></p> <p>31. Declare, implement, and access methods. <i>Examples: private, public, protected; method parameters; return type; void; return value; instance methods; static methods; overloading</i></p>	<p>30. Chapter 11</p> <p>31. Chapter 13, Lessons 3, 4, 5 Text Chapter 14, Lesson 3 Text Chapter 15, Lesson 2 Text</p>
<p style="text-align: center;">Compilation and Debugging</p>	<p>32. Troubleshoot syntax errors, logic errors, and runtime errors. <i>Examples: print statements, java command output, logic errors, console exceptions, stack trace evaluation</i></p> <p>33. Utilize debugging tools to suspend program execution and to examine, step through, and reset execution of code. <i>Examples: Visual Debugger, brute force method, backtracking, program slicing</i></p> <p>34. Utilize common error recovery strategies to detect errors and write a strategy to implement and handle the error. <i>Examples: statement mode, error productions</i></p>	<p>32. Chapter 9, Lesson 1 Text Chapter 10, Lesson 1 Text</p> <p>33. Chapter 10, Lesson 2 Text</p> <p>34. Chapter 9, Lessons 2, 3 Text Chapter 10, Lesson 1 Text Chapter 24, Lesson 3 Text</p>