CompuScholar, Inc.

Alignment to Utah

"Computer Programming 1" Course Standards (C# version)

Utah Course Details:

Course Name:	Computer Programming 1
Primary Cluster:	CTE / IT
Course Code(s):	35.02.00.00.030
Credit:	0.5 (First Semester)
Grade Level:	10th-12th
State Standards Link:	Computer Programming 1 Strands and Standards (August 2019)

CompuScholar Course Details:

Course Title:	Windows Programming with C#
Course ISBN:	978-0-9887070-0-9
Course Year:	2019

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.

Course Description

CompuScholar's "Windows Programming with C#" course is a 2-semseter experience that covers Utah's Computer Programming 1 and Computer Programming 2 standards. The course uses the Microsoft C# language to teach foundational coding skills.

Course Standards

STRAND 1: Students will be familiar with and use a programming language IDE (Integrated Development Environment).	**CITATION(S)**
Standard 1: Demonstrate concept knowledge of different languages.	
a. Describe the difference between an interpreted language vs a compiled language.	Chapter 2, Lesson 1
b. Identify characteristics of high-level and low-level languages.	Chapter 1, Lesson 4

Standard 2: Demonstrate the ability to use an IDE.	
a. Use an IDE to develop, compile, and run programs.	Chapter 2, Lessons 2-3
b. Understand the difference between syntax, run-time, and logic errors.	Chapter 10, Lesson 3
c. Use the debugger to identify errors.	Chapter 10, Lessons 1, 2, 4

STRAND 2: Students will understand program development methodology.	**CITATION(S)**
Standard 1: Demonstrate the ability to use good programming style.	
a. Demonstrate proper use of white space (between lines and indentation).	Chapter 2, Lesson 3
b. Use appropriate naming conventions for identifiers (variables, methods, functions, and file names).	Chapter 4, Lesson 2
c. Construct identifiers with meaningful format; camelCase and underscore	Chapter 4, Lesson 2
Standard 2: Understand the software development life-cycle.	
a. Identify specifications and understand requirements to create a solution to a problem.	Chapter 7, Lesson 3 Chapter 18, Lessons 1-2 Suppl. Chapter 2, Lesson 1
b. Develop a program using external documentation (flowcharts, abstracts, and pseudocode) to break down the problem into sub-components.	Chapter 7, Lesson 3 Chapter 12, Lesson 1 Chapter 18, Lessons 1-2
c. Design solutions using algorithms.	Chapter 7, Lesson 3 Chapter 18, Lessons 1-2
d. Write the code to implement the algorithm.	Chapter 7, Activity 1 Chapter 18, Lesson 3
e. Test program for verification of errors and proper functionality.	Chapter 10, Lessons 3-4 Chapter 18, Lesson 4
f. Provide internal comments in the IDE that explain functionality through documentation (i.e comments, notes, program instructions)	Chapter 2, Lesson 3
g. Redo all steps as needed.	Chapter 10, Lesson 4 Chapter 18, Lesson 4
Standard 3: Identify the components of a programming language syntax.	
a. Understand keywords, identifiers, operators, and operands.	Chapter 4, Lessons 1-2 Chapter 5, Lesson 1 Chapter 7, Lesson 1
b. Understand statements and expressions in a program.	Chapter 2, Lesson 3 Chapter 5, Lesson 1

c. Understand program components such as functions, methods, or procedures.	Chapter 2, Lesson 3 Chapters 9, 13

STRAND 3: Students will demonstrate effective use of commands and operations.	**CITATION(S)**
Standard 1: Employ basic use of elements and data types of a programming language.	
a. Declare, initialize, and assign values to constants and variables.	Chapter 4, Lesson 2
b. Demonstrate the ability to use input and output commands.	Chapter 2, Lessons 3-4 Chapter 3, Lessons 2-3 Chapter 6
c. Declare and use variable types (primitives, reference, or object).	Chapter 4, Lessons 1-4 Chapter 13, Lesson 1
d. Identify proper data types for a specified application (boolean, integer, floating point, strings).	Chapter 4, Lessons 1, 2, 4
Standard 2: Employ basic arithmetic expressions.	
a. Use basic arithmetic operators (modulus, multiplication, division, addition, subtraction).	Chapter 7, Lesson 1
b. Understand order of operation of expressions.	Chapter 5, Lesson 1
c. Write expressions that mix floating-point and integer expressions.	Chapter 4, Lesson 2

STRAND 4: Students will properly employ control and loop structures.	**CITATION(S)**
Standard 1: Demonstrate the ability to use relational and logical operators in programs.	
a. Compare values using relational operators (<, >, ==, >=, <=, etc.)	Chapter 5, Lesson 1
b. Form complex expressions using logical operators.	Chapter 5, Lesson 1
Standard 2: Demonstrate the ability to use decisions in programs.	
a. Employ simple IF structures.	Chapter 5, Lesson 2
b. Use IF-ELSE and nested IF-ELSE structures.	Chapter 5, Lesson 2 Chapter 17, Activity 3
Standard 3: Demonstrate the ability to use loops in programs.	
a. Demonstrate knowledge between for-loops, while-loops, and do-while loops.	Chapter 5, Lessons 3-4
b. Describe the various ways that loops can end (i.e., sentinel, break, condition fail, etc.).	Chapter 5, Lessons 3-4

c. Design loops so they iterate the correct number of times (i.e., off by one errors, infinite loops, etc.).	Chapter 5, Lessons 3-4
d. Utilize nested loops.	Chapter 5, Lesson 4 Chapter 14, Lessons 1, 3

STRAND 5: Students will be aware of career opportunities in the Computer Programming/Software Engineering industry and ethical applications.	**CITATION(S)**
Standard 1: Investigate career opportunities, trends, and requirements related to computer programming/software engineering careers.	
a. Identify the members of a computer programming/software engineering team: team leader, analyst, senior developer, junior developer, and client/subject matter expert.	Suppl. Chapter 2, Lesson 2
b. Describe work performed by each member of the computer programming/software engineering team.	Suppl. Chapter 2, Lesson 2
c. Investigate trends and traits associated with computer programming/software engineering careers (creativity, technical, leadership, collaborative, problem solving, design, etc.).	Suppl. Chapter 2, Lesson 2
d. Discuss related career pathways.	Suppl. Chapter 2, Lesson 2
Standard 2: Have an understanding of current ethical issues dealing with computer programming and information in society.	
a. Explain the impact software can have on society (i.e., privacy, piracy, copyright laws, ease of use, etc.).	Chapter 1, Lesson 5 Suppl. Chapter 3, Lessons 1-2
b. Explain the ethical reasons for creating reliable and robust software.	Chapter 1, Lesson 5
c. Describe how computer-controlled automation affects a workplace and society.	Suppl. Chapter 3, Lessons 2-3

STRAND 6: Students will employ arrays and strings. (Semester 2 Strands)	**CITATION(S)**
Standard 1: Demonstrate the ability to use arrays.	
a. Declare and initialize arrays.	Chapter 11, Lesson 1
b. Perform data input to and output from arrays.	Chapter 11, Lesson 1
c. Perform operations on arrays.	Chapter 11, Lesson 1
d. Iterate through the structure (i.e. for-each, enhanced for, or iterators)	Chapter 11, Lessons 1, 3
Standard 2: Demonstrate the ability to use strings in programs.	
a. Compare string values.	Chapter 4, Lesson 4 Chapter 8, Lesson 1

b. Find the length of a string.	Chapter 4, Lesson 4 Chapter 8, Lesson 1
c. Create and locate substrings.	Chapter 8, Lesson 1
d. Concatenate string values.	Chapter 4, Lesson 4

STRAND 7: Students will properly employ object-oriented programming techniques.	**CITATION(S)**
Standard 1: Demonstrate the ability to use existing classes.	
a. Instantiate objects.	Chapter 13, Lesson 1
b. Use an object's data members.	Chapter 13, Lesson 2
c. Use an object's member functions (methods).	Chapter 9 Chapter 13, Lesson 2
Standard 2: Demonstrate the ability to create user-defined classes.	
a. Create and use data members (instance variables).	Chapter 13, Lesson 2
b. Create constructors to initialize the data members.	Chapter 13, Lesson 4
c. Create and use member functions (methods).	Chapter 9 Chapter 13, Lesson 2
Standard 3: Demonstrate proper design principles with classes.	
a. Create classes that are well encapsulated (private data members).	Chapter 13, Lesson 3
b. Properly use modifiers and accessors (getters and setters).	Chapter 13, Lesson 3

STRAND 8: Students will apply programming skill as an effective member of a team demonstrating the ability to collaborate with others.	**CITATION(S)**
Standard 1: Demonstrate the ability to apply knowledge to a programming project.	
a. Formalize specifications.	Chapter 18, Lessons 1-2 Suppl. Chapter 2, Lesson 1
b. Choose proper input parameters.	Chapter 18, Lessons 1-2 Suppl. Chapter 2, Lesson 1
c. Choose relevant data structures and processing.	Chapter 18, Lessons 1-2 Suppl. Chapter 2, Lesson 1
d. Design relevant output.	Chapter 18, Lessons 1-2 Suppl. Chapter 2, Lesson 1

e. Use relevant test data.	Chapter 10, Lesson 4 Chapter 18, Lesson 4
f. Provide detailed documentation	Chapter 18, Lessons 1-2 Suppl. Chapter 2, Lesson 1, 4
Standard 2: Demonstrate the ability to use teamwork and collaboration in a programming project.	
a. Divide a project among programmers.	Chapter 18
b. Coordinate work with others in the group.	Chapter 18
c. Complete assigned work according to predetermined deadlines.	Chapter 18
d. Participate in a peer performance evaluation.	Chapter 18
e. Demonstrate professionalism in team relationships, communication, timeliness, and attitude.	Chapter 18 Suppl. Chapter 2, Lesson 2