CompuScholar, Inc.

Alignment to Utah

"Computer Programming 2" Course Standards (C# version)

Utah Course Details:

Course Name:	Computer Programming 2
Primary Cluster:	CTE / IT
Course Code(s):	35.02.00.00.032
Credit:	0.5 (Second Semester)
Grade Level:	10th-12th
State Standards Link:	Computer Programming 2 Strands and Standards (July 2018)

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 environment.	**CITATION(S)**
Standard 1: Demonstrate knowledge of software concepts.	
a. Identify software categories e.g. application software, web-based software, mobile application, or operating system.	Chapter 1, Lesson 3
b. Describe the difference between an interpreted language vs a compiled language.	Chapter 2, Lesson 1
Standard 2: Demonstrate the ability to compile, debug, and execute programs.	
a. Demonstrate how to use an editor/IDE to 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. Demonstrate how to debug programs.	Chapter 10, Lessons 1, 2, 4

STRAND 2: Students will employ accepted programming methodology.	**CITATION(S)**
Standard 1: Demonstrate the ability to use good programming style.	
a. Demonstrate how to use white space properly.	Chapter 2, Lesson 3
b. Employ an appropriate naming convention.	Chapter 4, Lesson 2
c. Construct identifiers with meaningful format (i.e.: camelCase, Underscores, and ALLCAPS).	Chapter 4, Lesson 2
Standard 2: Understand that software development is a process and use a variety of creation techniques to develop 21st Century Skills.	
a. Understand specifications and requirements for computer programs.	Chapter 7, Lesson 3 Chapter 18, Lessons 1 - 2 Suppl. Chapter 2, Lesson 1
b. 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 and other problem solving techniques.	Chapter 7, Lesson 3 Chapter 14 Chapter 18, Lessons 1 - 2
d. Write the code for a program.	Throughout the course
e. Test programs for errors and proper functionality.	Chapter 10, Lessons 3-4 Chapter 18, Lesson 4
f. Provide internal and external documentation for a program during development.	Chapter 2, Lesson 3 Chapter 7, Lesson 3 Chapter 18, Lessons 1 - 2
g. Redo all steps as needed.	Chapter 10, Lesson 4 Chapter 18, Lesson 4
Standard 3: Identify the syntactical components of a programming language.	
a. Identify keywords, identifiers, operators, and operands.	Chapter 4, Lessons 1-2 Chapter 5, Lesson 1 Chapter 7, Lesson 1
b. Identify the entry-point of a program.	Chapter 2, Lesson 3
c. Identify statements and expressions in a program.	Chapter 2, Lesson 3 Chapter 5, Lesson 1

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

STRAND 3: Students will properly use language-fundamental commands and operations.	**CITATION(S)**
Standard 1: Demonstrate the ability to use basic elements of a specific language.	
a. Write programs formatted based on the conventions of the utilized language.	Chapter 2, Lesson 3 and Throughout the course
b. Declare, initialize, and assign values to constants and variables.	Chapter 4, Lesson 2
c. Demonstrate the ability to use input and output commands.	Chapter 2, Lessons 3-4 Chapter 3, Lessons 2-3 Chapter 6
d. Communicate clearly with output values stored in identifiers.	Chapter 2, Lesson 4 Chapter 4, Lessons 2 - 4
e. Demonstrate the ability to use strings.	Chapter 4, Lesson 4 Chapter 8
Standard 2: Employ basic arithmetic expressions in programs.	
a. Use basic arithmetic operators (modulus, multiplication, division, addition, subtraction).	Chapter 4, Lesson 2
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
Standard 3: Demonstrate the ability to use data types in programs.	
a. Declare and use variable types (primitives, reference, or object).	Chapter 4, Lessons 2 - 4 Chapter 13, Lesson 1
b. Declare and use constants.	Chapter 4, Lesson 2
c. Know the difference between data types and their application (boolean, integer, floating point, strings).	Chapter 4, Lessons 1 - 4

STRAND 4: Students will properly employ control structures.	**CITATION(S)**
Standard 1: Demonstrate the ability to use relational and logical operators in programs.	
a. Compare values using relational operators.	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 structures.	Chapter 5, Lesson 2
c. Write programs with nested IF-ELSE structures.	Chapter 5, Lesson 2 Chapter 17, Activity 3
d. Make multiple-way selections (switch, case).* (Language specific)	N/A
Standard 3: Demonstrate the ability to use loops (iteration) in programs.	
a. Use initial (starting) value, terminal (ending) condition, and incrementation (change)in loops.	Chapter 5, Lessons 3 - 4
b. Construct pretest loops (while), posttest loops (do-while), and for loops.	Chapter 5, Lessons 3 - 4
c. Describe the various ways that loops can end (i.e., sentinel, break, condition fail, etc.).	Chapter 5, Lessons 3 - 4
d. Design loops so they iterate the correct number of times (i.e., off by one errors, infinite loops, etc.).	Chapter 5, Lessons 3 - 4
e. Accumulate running totals using loops.	Chapter 5, Lesson 4
f. Utilize nested loops.	Chapter 5, Lesson 4 Chapter 14, Lessons 1, 3
Standard 4: Demonstrate the ability to use modularity in programs using functions or methods.	
a. Demonstrate how to use language-defined functions and/or methods. *	Chapter 3, Lessons 2 - 3 Chapter 7, Lesson 2 Chapter 16, Lesson 4
b. Utilize value and/or reference parameters. *	Chapter 9, Lessons 2 - 3
c. Understand the scope of identifiers (local, global (class), and instance variables). *	Chapter 4, Lesson 2 Chapter 13, Lesson 2
d. Return values.	Chapter 9, Lessons 2 - 3

STRAND 5: Students will demonstrate knowledge of current ethical issues dealing with computers and information in a global society using 21st Century Skills.	**CITATION(S)**
Standard 1: Demonstrate knowledge of the social and ethical consequences of computers.	
a. Explain the ethical reasons for creating reliable and robust software.	Chapter 1, Lesson 5
b. 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

c. Show how security concerns can be addressed in an application (i.e., biometrics, passwords, information hiding, etc.).	Chapter 1, Lesson 6 Suppl. Chapter 3, Lesson 1
d. Describe how computer-controlled automation affects a workplace and society.	Suppl. Chapter 3, Lessons 2 - 3
e. Give examples of ways to protect information on computer systems (attacks, viruses, malware, etc.).	Chapter 1, Lesson 6

STRAND 6: Students will be aware of career opportunities in the Computer Programming/Software Engineering industry and of its history.	**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

STRAND 7: Students will employ static (array), dynamic (vector, ArrayList, etc.) list structures, and strings. (Semester 2 Strands)	**CITATION(S)**
Standard 1: Demonstrate the ability to use static arrays/lists in programs.	
a. Declare and initialize arrays/lists of all applicable types.	Chapter 11, Lesson 1
b. Perform data input to and output from arrays/lists.	Chapter 11, Lessons 1 - 2
c. Perform operations on arrays/lists including sort arrays.	Chapter 11 Chapter 14, Lessons 1, 3
d. Iterate through the structure (i.e. for-each, enhanced for, or iterators)	Chapter 11, Lesson 3
Standard 2: Demonstrate the ability to use dynamic arrays/lists (i.e. vectors, ArrayList, or generic lists)	
a. Declare and initialize a dynamic array/list.	Chapter 11, Lessons 1 - 2
b. Add and remove items from the array/list.	Chapter 11, Lessons 1 - 2
c. Output data from arrays/lists.	Chapter 11, Lessons 1 - 3
d. Perform operations on arrays/lists.	Chapter 11, Lessons 1 - 3

e. Iterate through the structure (i.e. for-each, enhanced for, or iterators)	Chapter 11, Lesson 3
Standard 3: 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. Copy part or all of string values into other strings.	Chapter 8, Lessons 1 - 2
d. Concatenate string values.	Chapter 4, Lesson 4
e. Locate substring positions.	Chapter 8, Lesson 1
f. Insert strings into other strings.	Chapter 8, Lesson 1

STRAND 8: 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 object data members (i.e., Java’s arr. length).	Chapter 11, Lesson 1 Chapter 13, Lesson 2
c. Use object member functions (methods).	Chapter 9 Chapter 13, Lesson 2 Chapter 16, Lessons 2 - 5
Standard 2: Demonstrate the ability to create user-defined classes.	
a. Create and use data members (instance variables).	Chapter 13, Lessons 2 - 3
b. Create a constructor to initialize the data members.	Chapter 13, Lesson 4
c. Create and use member functions (methods).	Chapter 9 Chapter 13, Lesson 2 Chapter 16, Lessons 2 - 5
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
c. Understand appropriate private and public modifiers according to program design.	Chapter 13, Lesson 3

STRAND 9: Students will properly use sequential files.	**CITATION(S)**
Standard 1: Demonstrate the ability to use sequential files in programs.	
a. Create and initialize sequential files.	Suppl. Chapter 1, Lesson 7
b. Store data to sequential files.	Suppl. Chapter 1, Lesson 7
c. Retrieve data from sequential files.	Suppl. Chapter 1, Lesson 7
d. Update sequential files.	Suppl. Chapter 1, Lesson 7

STRAND 10: Students will apply appropriate programming skill as an effective member of a team demonstrating the ability to collaborate with others (www.p21.org).	**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 appropriate data structures and processing.	Chapter 18, Lessons 1 - 2 Suppl. Chapter 2, Lesson 1
d. Design appropriate output.	Chapter 18, Lessons 1 - 2 Suppl. Chapter 2, Lesson 1
e. Use appropriate test data.	Chapter 10, Lesson 4 Chapter 18, Lesson 4
f. Write good documentation.	Chapter 18, Lessons 1 - 2 Suppl. Chapter 2, Lessons 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. Present work to a group.	Chapter 18
c. Coordinate work with others in the group.	Chapter 18
d. Complete assigned work according to predetermined deadlines.	Chapter 18
e. Participate in a peer performance evaluation.	Chapter 18
f. Demonstrate professionalism in team relationships, communication, timeliness, and attitude.	Chapter 18 Suppl. Chapter 2, Lesson 2