CompuScholar, Inc.

Alignment to Idaho **"Programming & Software Development"** Standards

Idaho Course Details:

Course Name:	Programming & Software Development
Primary Cluster:	CTE / Engineering & Technology Education
Credit:	1
Grade Level:	9th-12th
Program Standards Link:	Programming & Software Development Program Standards (2014 version)

CompuScholar Course

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#" is a computer science course based on the Microsoft C# language. Students will learn classic computer science concepts and introductory programming skills using the Visual Studio development environment.

Course Standards

CONTENT STANDARD 1.0: UNDERSTAND PROGRAMMING PRINCIPLES	CITATION(S)
Performance Standard 1.1: Demonstrate Critical Thinking and Problem-Solving Skills as they Apply to Programming	
1.1.1 Apply basic programming principles.	Chapter 2, Lesson 3
1.1.2 Describe and differentiate procedural and object-oriented programming.	Chapter 9, Lesson 1 Chapter 12, Lesson 1
1.1.3 Apply the features of object-oriented programming languages.	Chapter 13 Chapter 16
1.1.4 Write a program that produces output.	Chapter 2, Lesson 2 Chapter 8, Lesson 2
1.1.5 Select identifiers to use within programs.	Chapter 4, Lesson 2 Chapter 9, Lesson 1
1.1.6 Improve programs by adding comments.	Chapter 2, Lesson 3
1.1.7 Write and run a program.	Chapter 2, Lesson 3 Chapter 3, Lessons 1 – 3

CONTENT STANDARD 2.0: PROBLEM SOLVING THROUGH PROGRAMMING	CITATION(S)
Performance Standard 2.1: Demonstrate Ability to Use Variables, Data Types, and String Manipulation to Solve Computer Problems Programmatically	
2.1.1 Demonstrate the process of declaring variables.	Chapter 4, Lesson 2
2.1.2 Display variable values.	Chapter 4, Lesson 2
2.1.3 Apply integral data types.	Chapter 4, Lessons 1 - 2 Chapter 5, Lessons 2 - 3
2.1.4 Apply floating-point data types.	Chapter 4, Lessons 1 - 2 Chapter 7, Lesson 1 Chapter 9, Lesson 2
2.1.5 Apply arithmetic operators.	Chapter 7, Lesson 1
2.1.6 Apply Boolean data type.	Chapter 5, Lessons 1 - 2
2.1.7 Apply numeric type conversion.	Chapter 4, Lesson 2 Chapter 8, Lesson 3 Chapter 8 Activity
2.1.8 Apply char data type.	Chapter 4, Lessons 1, 2, 4 Chapter 8 Activity
2.1.9 Apply string data type.	Chapter 4, Lessons 3 – 4 Chapter 8
2.1.10 Define named constants and enumerations.	Chapter 4, Lesson 2 Chapter 16, Lesson 1

CONTENT STANDARD 3.0: USE LOGIC IN PROGRAMMING	CITATION(S)
Performance Standard 3.1: Demonstrate Effective Use of Selection Structures to Add Logic to Programs	
3.1.1 Demonstrate logic-planning tools and decision-making.	Chapter 5, Lessons 2 - 4 Chapter 7, Lesson 3
3.1.2 Make decision using the “if” statement.	Chapter 5, Lesson 2
3.1.3 Make decisions using the if-else statement.	Chapter 5, Lesson 2
3.1.4 Apply compound expressions in if statements.	Chapter 5, Lesson 1
3.1.5 Make decisions using the switch statement.	N/A
3.1.6 Apply the conditional operator.	Chapter 5, Lessons 1 - 2
3.1.7 Apply the NOT operator.	Chapter 5, Lesson 1

3.1.8. Describe how to avoid common errors when making decisions, and apply problem-solving skills in context.	Chapter 5, Lessons 1, 2, 4

CONTENT STANDARD 4: PROGRAMMING AND VALIDATION	**CITATION(S)**
Performance Standard: 4.1: Demonstrate Ability to Test, Debug and Validate Programming Applications	
4.1.1 Locate a logic error by stepping through the code.	Chapter 10, Lessons 1 – 2
4.1.2 Locate logic errors using breakpoints.	Chapter 10, Lesson 1
4.1.3 Fix syntax and logic errors.	Chapter 10, Lesson 4
4.1.4 Select appropriate test data for an application.	Chapter 10, Lesson 4

CONTENT STANDARD 5.0: UNDERSTAND REPETITION IN PROGRAMMING	**CITATION(S)**
Performance Standard 5.1: Differentiate Between the Various Types of Repetition	
5.1.1 Apply the loop structure.	Chapter 5, Lessons 3 – 4
5.1.2 Create loops using the while statement.	Chapter 5, Lesson 4
5.1.3 Create loops using the for statement.	Chapter 5, Lesson 3
5.1.4 Create loops using the do statement.	Chapter 5, Lesson 4
5.1.5 Apply nested loops.	Chapter 5, Lesson 4 Chapter 14, Lesson 1
5.1.6 Apply accumulators.	Chapter 7, Lessons 1, 4
5.1.7 Understand and describe how to improve loop performance	Chapter 14, Lessons 1, 3

CONTENT STANDARD 6.0: DEMONSTRATE PROGRAMMING FUNCTIONALITY	**CITATION(S)**
Performance Standard 6.1: Use Methods to Increase Functionality and to Modularize Programs	
6.1.1 Describe methods and implementation hiding.	Chapter 9, Lessons 1 – 3 Chapter 12, Lesson 1 Chapter 13, Lesson 3
6.1.2 Write methods with no parameters and no return value.	Chapter 9, Lesson 1
6.1.3 Write methods that require a single argument.	Chapter 9, Lesson 2

6.1.4 Write methods that require multiple arguments.	Chapter 9, Lesson 2
6.1.5 Write a method that returns a value.	Chapter 9, Lessons 2 - 3
6.1.6 Pass an array to a method.	Chapter 11, Lesson 1
6.1.7 Overload methods.	N/A
6.1.8 Demonstrate how to avoid methods.	N/A
6.1.9 Apply optional parameters.	N/A

CONTENT STANDARD 7.0: UNDERSTAND ARRAYS AND STRUCTURE CONCEPTS	**CITATION(S)**
Performance Standard 7.1: Demonstrate Understanding of Arrays and Structure and Apply Concepts In Program Development	
7.1.1 Declare an array and assign values to array elements.	Chapter 11, Lesson 1
7.1.2 Access array elements.	Chapter 11, Lesson 1
7.1.3 Search an array using a loop.	Chapter 11, Lesson 1
7.1.4 Apply multidimensional arrays.	Chapter 11, Lesson 1

CONTENT STANDARD 8.0: UNDERSTAND CLASSES IN PROGRAMMING	**CITATION(S)**
Performance Standard 8.1: Students will demonstrate understanding of Object-Oriented Programming Concepts	
8.1.1 Describe and apply class concepts.	Chapter 12 Chapter 13
8.1.2 Create classes from which objects can be instantiated.	Chapter 13, Lesson 1
8.1.3 Create objects.	Chapter 13, Lessons 1 - 2
8.1.4 Create properties, including auto-implemented properties.	Chapter 13, Lesson 2
8.1.5 Use public fields and private methods.	Chapter 13, Lesson 3
8.1.6 Define the "this" reference.	N/A
8.1.7 Write constructors.	Chapter 13, Lesson 4
8.1.8 Use object initializers.	Chapter 13, Lesson 4

8.1.9 Overload operators.	N/A
8.1.10 Declare an array of objects.	Chapter 11, Lesson 1
8.1.11 Use sorting methods with an array of objects.	Chapter 14, Lessons 1, 3
8.1.12 Write destructors.	N/A (not available in C#)
8.1.13 Describe and demonstrate inheritance.	Chapters 16 and 17
8.1.14 Extend classes.	Chapter 16, Lesson 1
8.1.15 Override base class methods.	Chapter 16, Lesson 3
8.1.16 Describe how a derived class object "is an" instance of the base class.	Chapter 12, Lesson 3 Chapter 16, Lesson 3
8.1.17 Define the object class.	Chapter 16, Lesson 4
8.1.18 Use base class constructors.	Chapter 16, Lesson 5
8.1.19 Create abstract classes.	Chapter 16, Lesson 1
8.1.20 Create use interfaces.	N/A
8.1.21 Apply extension methods.	N/A
8.1.22 Describe the benefits of inheritance.	Chapter 12, Lessons 1, 3 Chapter 16, Lesson 1
8.1.23 Recognize inheritance in GUI applications.	N/A

CONTENT STANDARD 9.0: UNDERSTAND PROGRAMMING AND EXCEPTIONS	**CITATION(S)**
Performance Standard 9.1: Demonstrate Exception-Handling in Program Development	
9.1.1 Compare and demonstrate traditional and object-oriented error-handling methods.	Chapter 10, Lesson 3
9.1.2 Cast data types.	N/A in the context of exceptions, though casting is described in Chapter 4, Lesson 2 and used elsewhere.
9.1.3 Catch multiple exceptions.	N/A
9.1.4 Apply the finally block.	N/A
9.1.5 Handle exceptions thrown from outside methods.	Chapter 10, Lesson 3 Supplemental Chapter 1, Lesson 7

9.1.6 Trace exceptions through the call stack.	N/A
9.1.7 Create exception classes.	N/A
9.1.8 Re-throw exceptions.	N/A

CONTENT STANDARD 10.0: UNDERSTAND PROGRAMMING AND EVENTS	**CITATION(S)**
Performance Standard 10.1: Use Event Handlers in Programs	
10.1.1 Define and apply event handling.	Chapter 3, Lesson 3 Chapter 6, Lesson 1
10.1.2 Define and describe delegates.	N/A
10.1.3 Declare own events and handlers.	N/A
10.1.4 Use built-in event handlers.	Chapter 3, Lesson 3 Chapter 6, Lesson 1
10.1.5 Handle control component events.	Chapter 3, Lesson 3 Chapter 6, Lesson 1
10.1.6 Handle mouse and keyboard events.	N/A
10.1.7 Manage multiple controls	Chapter 6 Chapter 6 Activity
10.1.8 Explain how to find more information on controls and events	Chapter 3, Lesson 4 Chapter 6, Lesson 1

CONTENT STANDARD 11.0: SYSTEMS PLANNING AND DEVELOPMENT	**CITATION(S)**
Performance Standards 11.1: Apply Concepts and Principles of Systems Planning and Development	
11.1.1 Describe the information systems development life cycle (SDLC).	Chapter 18 Supplemental Chapter 2, Lesson 1
11.1.2 Discuss how to evaluate off-the-shelf software.	N/A
11.1.3 Explain reuse and its role in software development.	Chapter 12, Lesson 1
11.1.4 Describe the skills required to be an effective project manager.	Supplemental Chapter 2, Lesson 2
11.1.5 List and describe the skill and activities of a project manager during project initiation, planning, execution, and closedown.	Supplemental Chapter 2, Lesson 1
11.1.6 Describe the steps for identifying and selecting projects and initiating and planning projects.	Supplemental Chapter 2, Lesson 1
11.1.7 Explain the need for and contents of a project scope statement.	Supplemental Chapter 2, Lesson 1

11.1.8 Compare various methods for assessing project feasibility.	N/A

CONTENT STANDARD 12.0: SYSTEMS ANALYSIS	CITATION(S)
Performance Standards 12.1: Demonstrate Competency with Systems Analysis Tools and Concepts	
12.1.1 Compare options for designing and conducting interviews to determine system requirements.	N/A
12.1.2 Develop a plan for conducting an interview to determine system requirements.	N/A
12.1.3 Explain the advantages and pitfalls of observing workers and analyzing business documents to determine system requirements.	N/A
12.1.4 Plan a joint application design session.	N/A
12.1.5 Use prototyping during requirements determination.	N/A
12.1.6 Select appropriate methods to elicit system requirements.	N/A
12.1.7 Describe how requirements determination techniques apply to development of Internet applications.	N/A
12.1.8 Demonstrate the logical modeling of processes through studying examples of data-flow diagrams, pseudo code, and flowcharts.	Chapter 7, Lesson 3

CONTENT STANDARD 13.0: PRINCIPLES OF DESIGN	CITATION(S)
Performance Standards 13.1: Demonstrate Knowledge Of Application Design Principles	
13.1.1 Explain the process of designing interfaces and dialogues and the deliverables for their creation.	N/A
13.1.2 Apply the general guidelines for interface design, including guidelines for layout design, structuring data-entry fields, providing feedback, and system help.	N/A
13.1.3 Concisely define each of the following key database design terms: relation, primary key, functional dependency, foreign key, referential integrity, field, data type, null value, demoralization, file organization, index, and secondary key.	N/A
13.1.4 Explain the role of designing databases in the analysis and design of an information system.	N/A
13.1.5 Transform an entity-relation (E-R) diagram into an equivalent set of well-structured (normalized) relations.	N/A
13.1.6 Merge normalized relations from separate user views into a consolidated set of well-structured relations.	N/A
13.1.7 Choose storage formats for fields in database tables.	N/A

13.1.8 Translate well-structured relations into efficient database tables.	N/A
13.1.9 Explain when to use different types of file organizations to store computer files.	N/A
13.1.10 Describe the purpose indexes and the important considerations in selecting attributes to be indexed.	N/A

CONTENT STANDARD 14.0: IMPLEMENTATION AND SUPPORT	**CITATION(S)**
Performance Standards 14.1: Demonstrate Knowledge of Application Implementation and Identify the Need for Ongoing Application Support	
14.1.1 Describe the process of coding, testing, and converting an organizational information system.	Chapter 18 Supplemental Chapter 2, Lesson 1
14.1.2 Outline the deliverables and outcomes of an organizational information system.	Chapter 18 Supplemental Chapter 2, Lesson 1
14.1.3 List the deliverables for documenting the system and for training and supporting users.	Chapter 18 Supplemental Chapter 2, Lesson 1
14.1.4 Compare the many modes available for organizational information system training, including self-training and electronic performance support systems.	N/A
14.1.5 Discuss the issues of providing support for end users.	Supplemental Chapter 2, Lesson 2
14.1.6 Explain why application implementation sometimes fails.	N/A
14.1.7 Describe several factors that influence the cost of maintaining an application.	N/A