CompuScholar, Inc.

Correlations to the Nevada Academic Content Standards (NVACS):
Computer Science, Grades 9 - 12

"Windows Programming with C#"

Nevada Course Details:

Content Area	Computer Science
Grade Range	9 - 12
Standards Document	Nevada Academic Content Standards for CS and IT (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.

Note 3: Citation(s) to “Supplemental” lessons or chapters can be found in Supplemental chapters at the end of each course.

Concepts

Algorithms and Programming

Indicator and Standard	**Citation(s)**
9-12.AP.A.1 - Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	Chapter 7, Lesson 3 Chapters 9, 17, 18
9-12.AP.V.1 - Demonstrate the use of both linked lists and arrays to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	Chapter 11
9-12.AP.C.1 - Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.	Chapter 5, Lessons 2 - 4 Chapter 11, Lesson 3
9-12.AP.C.2 - Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	Chapters 16, 17, 18 Supplemental Chapter 1, Lessons 5 - 6

9-12.AP.M.1 - Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.	Chapters 9, 12, 13, 16, 18
9-12.AP.M.2 - Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	Chapters 9, 11, 12, 13, 16, 18
9-12.AP.PD.1 - Systematically design and develop programs for broad audiences by incorporating feedback from users.	Chapter 18 Supplemental Chapter 2, Lesson 1
9-12.AP.PD.2 - Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.	Chapter 1, Lesson 5
9-12.AP.PD.3 - Evaluate and refine computational artifacts to make them more usable by all and accessible to people with	N/A
9-12.AP.PD.4 - Design and develop computational artifacts working in team roles using collaborative tools.	Chapter 18 Supplemental Chapter 2, Lesson 1
9-12.AP.PD.5 - Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	Chapter 7, Lesson 3 Chapter 18 Supplemental Chapter 1, Lesson 6 Supplemental Chapter 2, Lesson 1

Computing Systems	
Indicator and Standard	**Citation(s)**
9-12.CS.D.1 - Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	Chapter 1, Lesson 2
9-12.CS.HS.1 - Compare levels of abstraction and interactions between application software, system software, and hardware layers.	Chapter 1, Lessons 2 - 4
9-12.CS.T.1 - Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	Chapter 10, Lesson 4 Chapter 18, Lesson 4 Supplemental Chapter 2, Lesson 1

Data and Analysis	
Indicator and Standard	**Citation(s)**
9-12.DA.S.1 - Translate between different bit representations of real-world phenomena, such as characters, numbers, and images (e.g., convert hexadecimal colors to decimal percentages, ASCII/Unicode representation).	Chapter 4, Lesson 5 Chapter 8 Activity Supplemental Chapter 1, Lesson 1
9-12.DA.S.2 - Evaluate the tradeoffs in how data elements are organized and where data is stored.	Chapter 4, Lessons 1 - 3 Chapter 11, Lessons 1 - 2 Supplemental Chapter 1, Lesson 8
9-12.DA.CVT.1 - Create interactive data visualizations or alternative representations using software tools to help others better understand real-world phenomena.	Chapters 12 and 13 Activities Chapter 18 Supplemental Chapter 1, Lessons 5 - 6

9-12.DA.IM.1 - Create computational models that represent the relationships among different elements of data collected from a phenomenon, process, or model.	Chapters 12 and 13 Activities Chapter 18 Supplemental Chapter 1, Lessons 5 - 6

Impacts of Computing	
Indicator and Standard	**Citation(s)**
9-12.IC.C.1 - Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	Chapter 1, Lesson 5 Supplemental Chapter 3
9-12.IC.C.2 - Test and refine computational artifacts to reduce bias and equity deficits.	N/A
9-12.IC.C.3 - Demonstrate ways a given algorithm applies to problems across disciplines.	Chapter 7, Lesson 3 Chapters 17, 18 Supplemental Chapter 3, Lesson 3
9-12.IC.C.4 - Explain the potential impacts of artificial intelligence on society.	Supplemental Chapter 3, Lesson 3
9-12.IC.SI.1 - Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.	Chapter 18 Supplemental Chapter 3, Lesson 4
9-12.IC.SLE.1 - Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	Chapter 1, Lesson 5
9-12.IC.SLE.2 - Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.	Chapter 1, Lesson 6 Supplemental Chapter 3, Lesson 1
9-12.IC.SLE.3 - Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.	Chapter 1, Lessons 5 - 6 Supplemental Chapter 3, Lesson 1

Networks and the Internet	
Indicator and Standard	**Citation(s)**
9-12.NI.NCO.1 - Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.	Supplemental Chapter 1, Lesson 2 Supplemental Chapter 1, Lesson 3
9-12.NI.C.1 - Give examples to illustrate how sensitive data can be affected by malware and other attacks.	Chapter 1, Lesson 6 Supplemental Chapter 3, Lesson 1
9-12.NI.C.2 - Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	Chapter 1, Lesson 6 Supplemental Chapter 3, Lesson 1
9-12.NI.C.3 - Compare various security measures, considering tradeoffs between the usability and security of a computing system.	Chapter 1, Lesson 6 Supplemental Chapter 3, Lesson 1
9-12.NI.C.4 - Explain tradeoffs when selecting and implementing cybersecurity recommendations.	Chapter 1, Lesson 6 Supplemental Chapter 3, Lesson 1

Practices

Practice 1. Fostering an Inclusive Computing Culture	
Description	**Citation(s)**
1. Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.	Chapter 18 Supplemental Chapter 2, Lesson 1
2. Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability.	Chapter 18 Supplemental Chapter 2, Lesson 1
3. Employ self- and peer-advocacy to address bias in interactions, product design, and development methods.	Chapter 18 Supplemental Chapter 2, Lesson 1

Practice 2. Collaborating Around Computing	
Description	**Citation(s)**
1. Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities.	Chapter 18
2. Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness.	Chapter 18
3. Solicit and incorporate feedback from, and provide constructive feedback to, team members and other	Chapter 18
4. Evaluate and select technological tools that can be used to collaborate on a project.	Chapter 18 Supplemental Chapter 3, Lesson 4

Practice 3. Recognizing and Defining Computational Problems	
Description	**Citation(s)**
1. Identify complex, interdisciplinary, real-world problems that can be solved computationally.	Chapter 7, Lesson 3 Chapter 14 Supplemental Chapter 1, Lessons 5 - 6 Supplemental Chapter 3, Lesson 3
2. Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.	Chapter 7, Lesson 3 Chapters 14, 16, 17 Supplemental Chapter 1, Lessons 5 - 6 Supplemental Chapter 3, Lesson 3
3. Evaluate whether it is appropriate and feasible to solve a problem computationally.	Supplemental Chapter 3, Lesson 3

Practice 4. Developing and Using Abstractions	
Description	**Citation(s)**
1. Extract common features from a set of interrelated processes or complex phenomena.	Chapter 9, Lesson 1 Chapter 12, Lessons 1, 3 Chapter 16, Lesson 1

2. Evaluate existing technological functionalities and incorporate them into new designs.	Chapter 3, Lesson 4 Objects from the .NET Framework are incorporated into programs throughout the course
3. Create modules and develop points of interaction that can apply to multiple situations and reduce complexity.	Chapters 9, 12, 13, 16
4. Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.	Supplemental Chapter 1, Lessons 5 - 6

Practice 5. Creating Computational Artifacts	
Description	**Citation(s)**
1. Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.	Chapter 18 Supplemental Chapter 2, Lesson 1
2. Create a computational artifact for practical intent, personal expression, or to address a societal issue.	Chapters 16, 17, 18
3. Modify an existing artifact to improve or customize it.	Activities in Chapters 13, 14, 16, 17, etc.

Practice 6. Testing and Refining Computational Artifacts	
Description	**Citation(s)**
1. Systematically test computational artifacts by considering all scenarios and using test cases.	Chapter 10, Lesson 4 Chapter 18, Lesson 4
2. Identify and fix errors using a systematic process.	Chapter 10, Lesson 4 Chapter 18, Lesson 4
3. Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.	Chapter 10, Lesson 4 Chapter 18, Lesson 4

Practice 7. Communicating About Computing	
Description	**Citation(s)**
1. Select, organize, and interpret large data sets from multiple sources to support a claim.	Chapter 18 (if desired) Supplemental Chapter 1, Lessons 5 - 6
2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.	Chapter 7, Lesson 3 Chapter 18, Lessons 1 - 2 Supplemental Chapter 2, Lesson 1
3. Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.	Chapter 1, Lesson 5

Advanced* Concepts

* These Advanced Computer Science Standards are considered to be higher level concepts that may be used by your advanced students, incorporated into upper level courses, and/or used in Career and Technical Education (CTE) programs.

Algorithms and Programming	
Indicator and Standard	**Citation(s)**
A9-12.AP.A.1 - Describe how artificial intelligence drives many software and physical systems.	Supplemental Chapter 3, Lesson 3
A9-12.AP.A.2 - Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.	Chapter 18 (Team project may have an AI focus if desired) Supplemental Chapter 3, Lesson 3
A9-12.AP.A.3 - Use and adapt classic algorithms to solve computational problems.	Chapter 7, Lesson 3 Chapter 14
A9-12.AP.A.4 - Evaluate algorithms in terms of their efficiency, correctness, and clarity.	Chapter 7, Lesson 3 Chapter 14
A9-12.AP.V.1 - Compare and contrast fundamental data structures and their uses.	Chapter 11 Supplemental Chapter 1, Lesson 8
A9-12.AP.C.1 - Illustrate the flow of execution of a recursive algorithm.	Chapter 14, Lessons 2 - 3
A9-12.AP.M.1 - Construct solutions to problems using student-created components, such as procedures, modules and/or objects.	Chapters 9, 12, 13, 16
A9-12.AP.M.2 - Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.	Chapter 14 Chapter 16
A9-12.AP.M.3 - Demonstrate code reuse by creating programming solutions using libraries and APIs.	Chapter 2, Lesson 1 .NET Framework used throughout (e.g. Chapter 7, Lesson 2, Chapter 11, Lesson 2)
A9-12.AP.PD.1 - Plan and develop programs for broad audiences using a software life cycle process.	Chapter 18 Supplemental Chapter 2, Lesson 1
A9-12.AP.PD.2 - Explain security issues that might lead to compromised computer programs.	Chapter 1, Lesson 6
A9-12.AP.PD.3 - Develop programs for multiple computing platforms.	Chapter 1, Lesson 4 Chapter 2, Lesson 1
A9-12.AP.PD.4 - Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.	Chapter 2, Lessons 1-2 Chapter 18 Supplemental Chapter 3, Lesson 4
A9-12.AP.PD.5 - Develop and use a series of test cases to verify that a program performs according to its design specifications.	Chapter 10, Lesson 4 Chapter 18, Lesson 4 Supplemental Chapter 2, Lesson 1

A9-12.AP.PD.6 - Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).	Adding to existing projects: Activities in Chapters 13, 14, 16, 17, etc. Iterative & regression testing: Chapter 10, Lesson 4 Chapter 18, Lesson 4 Supplemental Chapter 2, Lesson 1
A9-12.AP.PD.7 - Evaluate key qualities of a program through a process such as a code review.	Chapter 10, Lesson 4 Chapter 18, Lesson 4 Supplemental Chapter 2, Lesson 1
A9-12.AP.PD.8 - Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.	Chapter 1, Lesson 4

Computing Systems	
Indicator and Standard	**Citation(s)**
A9-12.CS.HS.1 - Categorize the roles of operating system software.	Chapter 1, Lesson 3
A9-12.CS.T.1 - Illustrate ways computing systems implement logic, input, and output through hardware components.	Chapter 1, Lesson 2

Data and Analysis	
Indicator and Standard	**Citation(s)**
A9-12.DA.CVT.1 - Use data analysis tools and techniques to identify patterns in data representing complex systems.	Supplemental Chapter 1, Lesson 5 Supplemental Chapter 1, Lesson 6
A9-12.DA.CVT.2 - Select data collection tools and techniques to generate data sets that support a claim or communicate information.	Supplemental Chapter 1, Lesson 5 Supplemental Chapter 1, Lesson 6
A9-12.DA.IM.1 - Evaluate the ability of models and simulations to test and support the refinement of hypotheses.	Supplemental Chapter 1, Lesson 5 Supplemental Chapter 1, Lesson 6

Impacts of Computing	
Indicator and Standard	**Citation(s)**
A9-12.IC.C.1 - Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.	Supplemental Chapter 3, Lesson 2 Supplemental Chapter 3, Lesson 3
A9-12.IC.C.2 - Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.	N/A
A9-12.IC.C.3 - Predict how computational innovations that have revolutionized aspects of our culture might evolve.	Supplemental Chapter 3, Lesson 2
A9-12.IC.SLE.1 - Debate laws and regulations that impact the development and use of software.	Chapter 1, Lessons 5 - 6 Supplemental Chapter 3, Lesson 3 Activity

Networks and the Internet	
Indicator and Standard	**Citation(s)**
A9-12.NI.NCO.1 - Describe the issues that impact network functionality (e.g., bandwidth, load, delay, and topology).	Supplemental Chapter 1, Lesson 2 Supplemental Chapter 1, Lesson 3
A9-12.NI.C.1 - Compare ways software developers protect devices and information from unauthorized access.	Chapter 1, Lesson 6 Supplemental Chapter 3, Lesson 1