

# ***Advanced Computer Science Program of Study and Complementary Course Standards***



This document was prepared by:

Office of Career Readiness, Adult Learning, and Education Options  
Nevada Department of Education  
755 N. Roop Street, Suite 201  
Carson City, NV 89701

[www.doe.nv.gov](http://www.doe.nv.gov)

**Draft for Review** by the Nevada State Board of Education on  
**July 26, 2023**

The Nevada Department of Education does not discriminate on the basis of race, color, religion, national origin, sex, disability, sexual orientation, gender identity or expression, or age in its programs and activities and provides equal access to the Boy Scouts and other designated youth groups.

For inquiries, contact the Equity Coordinator at (775) 687-9200.

**Nevada State Board of Education**

Joseph Arrascada  
Dr. René Cantú  
Maggie Carlton  
Katie Coombs  
Dr. Katherine Dockweiler, Vice President  
Tamara Hudson  
Tim Hughes  
Michael Keyes  
Felicia Ortiz, President  
Dr. Summer Stephens  
Mike Walker

**Nevada Department of Education**

Jhone M. Ebert  
Superintendent of Public Instruction

Craig Statucki  
Interim Deputy Superintendent for Educator Effectiveness and Family Engagement

Cindi Chang  
Interim Director for the Office of Career Readiness, Adult Learning, and Education Options

Denise Burton  
Education Programs Professional, Office of Career Readiness, Adult Learning, and Education Options

**Vision**

*All Nevada students are equipped and feel empowered to attain their vision of success*

**Mission**

*To improve student achievement and educator effectiveness by ensuring opportunities, facilitating learning, and promoting excellence*



---

**Table of Contents**

Acknowledgements / Standards Development Members / Business and Industry Validation.....vii

Introduction .....ix

Program Information ..... 1

Content Standard 1.0 Integrate Career and Technical Student Organizations (CTSOs) ..... 2

Content Standard 2.0 Understand Algorithms and Programming..... 3

Content Standard 3.0 Understand Computing Systems ..... 5

Content Standard 4.0 Understand Data and Analysis..... 6

Content Standard 5.0 Understand Impacts of Computing..... 7

Content Standard 6.0 Understand Networks and the Internet ..... 8

Content Standard 7.0 Understand Emerging Technologies..... 9

Complementary Courses..... 11

Software and App Development..... 13

## Acknowledgements

The development of Nevada career and technical education (CTE) standards and assessments is a collaborative effort sponsored by the Nevada Department of Education (NDE) Office of Career Readiness, Adult Learning, and Education Options. The Nevada Department of Education relies on educators and industry representatives who have the technical expertise and teaching experience to develop standards and performance indicators that truly measure student skill attainment. More importantly, the NDE would like to recognize the time and commitment by the writing team members in developing the career and technical standards for Advanced Computer Science.

## Standards Development Members

Name	Occupation/Title	Stakeholder Affiliation	School/Organization
Fran Bromley-Norwood	Administrator	District Administrator	Clark County School District
Brian Johnson	Instructor	Secondary Educator	Doral Academy Red Rock High School, State Public Charter School Authority
Lata Koneru	Instructor	Postsecondary Educator	College of Southern Nevada, Las Vegas
Lloyd Mann	Instructor	Secondary Educator	Shadow Ridge High School, Clark County School District
Kathryn Milliken	Instructor	Secondary Educator	Damonte Ranch High School, Washoe County School District
Mark Newburn	Owner	Business and Industry Representative	Vizics Inc., Las Vegas
Robbie Pearce	Instructor	Secondary Educator	Cheyenne High School, Clark County School District

## Business and Industry Validation

All CTE standards developed through the Nevada Department of Education are validated by business and industry through one or more of the following processes: (1) the standards are developed by a team consisting of business and industry representatives, or (2) a separate review panel is coordinated with industry experts to ensure the standards include the proper content, or (3) nationally recognized standards currently endorsed by business and industry.

The Advanced Computer Science standards were validated through active participation of business and industry representatives on the development team.

---

## Introduction

The standards in this document are designed to clearly state what the student should know and be able to do upon completion of a high school Advanced Computer Science program of study. These standards are designed for a two-credit course sequence that prepares the student for a technical assessment directly aligned to the standards.

These exit-level standards are designed for the student to complete all standards through their completion of a program of study. These standards are intended to guide curriculum objectives for a program of study.

The standards are organized as follows:

- **Content Standards** are general statements that identify major areas of knowledge, understanding, and the skills students are expected to learn in key subject and career areas by the end of the program.
- **Performance Standards** follow each content standard. Performance standards identify the more specific components of each content standard and define the expected abilities of students within each content standard.
- **Performance Indicators** are very specific criteria statements for determining whether a student meets the performance standard. Performance indicators may also be used as learning outcomes, which teachers can identify as they plan their program learning objectives. The indicators are followed by designations that reflect the course sequence (e.g., 12 for the first-year course of a two-year program and 22 for the second-year course, C is to designate the indicators to be taught in the complementary courses) as referenced in the Core Course Sequence table.

The crosswalks and alignments are located in the Program Supplemental Program Resources document. These will show where the performance indicators support the Nevada Academic Content Standards. For individual course descriptions, please reference the Supplemental Program Resource or the Nevada CTE Catalog.

All students are encouraged to participate in the career and technical student organization (CTSO) that relates to the Advanced Computer Science program. CTSOs are co-curricular national organizations that directly reinforce learning in the CTE classroom through curriculum resources, competitive events, and leadership development. CTSOs provide students the ability to apply academic and technical knowledge, develop communication and teamwork skills, and cultivate leadership skills to ensure college and career readiness.

The Employability Skills for Career Readiness identify the skills needed to be successful in all careers and must be taught as an integrated component of all CTE course sequences. These standards are available in a separate document.

The **Standards Reference Code** is only used to identify or align performance indicators listed in the standards to daily lesson plans, curriculum documents, or national standards. The Standards Reference Code is an abbreviated name for the program, and the content standard, performance standard and performance indicator are referenced in the program standards. This abbreviated code for identifying standards uses each of these items. For example, ADVCS is the Standards Reference Code for Advanced Computer Science. For Content Standard 2, Performance Standard 3 and Performance Indicator 4 the Standards Reference Code would be ADVCS.2.3.4.

## Advanced Computer Science

### Program Information

- Program of Study: **Advanced Computer Science**
- Standards Reference Code: **ADVCS**
- Career Cluster: **Information Technology**
- Career Pathway(s): **Programming and Software Development**
- Program Length: **2-year, completed sequentially**
- CTSO: **SkillsUSA / FBLA**

### Program Structure Required Program of Study Courses

The core course sequencing is provided in the following table. Complementary Courses are available and provided later in this document. The following courses provide a completed program of study. The Lab is a complementary course available concurrently with the Advanced Computer Science II course.

**Core Course Sequence (R) with Lab Course(s) (C)**

Required/ Complementary	Course Title	Abbreviated Name
R	Advanced Computer Science I	ADV COMP SCI I
R	Advanced Computer Science II	ADV COMP SCI II
C	Advanced Computer Science II LAB	ADV COMP SCI II L

**CONTENT STANDARD 1.0: INTEGRATE CAREER AND TECHNICAL STUDENT ORGANIZATIONS (CTSOs)****Performance Standard 1.1: Explore the History and Organization of CTSOs**

- 1.1.1 Discuss the requirements of CTSO participation/involvement as described in Carl D. Perkins Law (12, 22, C)
- 1.1.2 Research nationally recognized CTSOs (12, 22, C)
- 1.1.3 Investigate the impact of federal and state government regarding the progression and operation of CTSOs (e.g., Federal Statutes and Regulations, Nevada Administrative Code [NAC], Nevada Revised Statutes [NRS]) (12, 22, C)

**Performance Standard 1.2: Develop Leadership Skills**

- 1.2.1 Discuss the purpose of parliamentary procedure (12, 22, C)
- 1.2.2 Demonstrate the proper use of parliamentary procedure (12, 22, C)
- 1.2.3 Differentiate between an office and a committee (12, 22, C)
- 1.2.4 Discuss the importance of participation in local, regional, state, and national conferences, events, and competitions (12, 22, C)
- 1.2.5 Participate in local, regional, state, or national conferences, events, or competitions (12, 22, C)
- 1.2.6 Describe the importance of a constitution and bylaws to the operation of a CTSO chapter (12, 22, C)

**Performance Standard 1.3: Participate in Community Service**

- 1.3.1 Explore opportunities in community service-related work-based learning (WBL) (12, 22, C)
- 1.3.2 Participate in a service learning (program related) and/or community service project or activity (12, 22, C)
- 1.3.3 Engage with business and industry partners for community service (12, 22, C)

**Performance Standard 1.4: Develop Professional and Career Skills**

- 1.4.1 Demonstrate college and career readiness (e.g., applications, resumes, interview skills, presentation skills) (12, 22, C)
- 1.4.2 Describe the appropriate professional/workplace attire and its importance (12, 22, C)
- 1.4.3 Investigate industry-standard credentials/certifications available within this Career Cluster™ (12, 22, C)
- 1.4.4 Participate in authentic contextualized instructional activities (12, 22, C)
- 1.4.5 Demonstrate technical skills in various student organization activities/events (12, 22, C)

**Performance Standard 1.5: Understand the Relevance of Career and Technical Education (CTE)**

- 1.5.1 Make a connection between program standards to career pathway(s) (12, 22, C)
- 1.5.2 Explain the importance of participation and completion of a program of study (12, 22, C)
- 1.5.3 Promote community awareness of local student organizations associated with CTE programs (12, 22, C)

---

**CONTENT STANDARD 2.0: UNDERSTAND ALGORITHMS AND PROGRAMMING****Performance Standard 2.1: Apply Algorithms**

- 2.1.1 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests (12)
- 2.1.2 Describe how artificial intelligence drives many software and physical systems (12)
- 2.1.3 Implement an artificial intelligence algorithm (e.g., machine learning) to play a game against a human opponent or solve a problem (22)
- 2.1.4 Use and adapt classic algorithms to solve computational problems (12)
- 2.1.5 Develop classic algorithms in code to solve computational problems (22)
- 2.1.6 Evaluate algorithms in terms of their efficiency, correctness, and clarity (22)

**Performance Standard 2.2: Implement Controls**

- 2.2.1 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made (12)
- 2.2.2 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions (12)
- 2.2.3 Illustrate the flow of execution of a recursive algorithm (22)
- 2.2.4 Implement conditional controls in code (22)
- 2.2.5 Implement recursive algorithms in code (22)

**Performance Standard 2.3: Utilize Variables**

- 2.3.1 Demonstrate the use of both LinkedLists and ArrayLists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables (12)
- 2.3.2 Compare and contrast fundamental data structures and their uses (12)
- 2.3.3 Implement arrays in code (12)
- 2.3.4 Implement ArrayLists and LinkedLists in code (12)
- 2.3.5 Implement type-safe variables (12)

**Performance Standard 2.4: Construct Solutions Using Modularity**

- 2.4.1 Decompose problems into smaller components through systematic analysis using constructs such as procedures, modules, and/or objects (12)
- 2.4.2 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs (12)
- 2.4.3 Construct solutions to problems using student-created components such as procedures, modules, and/or objects (22)
- 2.4.4 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution (22)
- 2.4.5 Demonstrate code reuse by creating programming solutions using libraries and APIs (22)

**Performance Standard 2.5: Demonstrate Programming and Development**

- 2.5.1 Systematically design and develop programs for broad audiences by incorporating feedback from users (12)
- 2.5.2 Evaluate software licenses that limit or restrict the use of computational artifacts when using resources such as libraries (12)
- 2.5.3 Evaluate and refine computational artifacts to make them more usable by all and accessible to people with disabilities (12)
- 2.5.4 Design and develop computational artifacts while working in team roles and using collaborative tools (12)
- 2.5.5 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs (12)
- 2.5.6 Plan and develop programs for broad audiences using a software life cycle process (e.g., requirements, design, coding, testing, and deployment) (22)
- 2.5.7 Explain security issues that might lead to compromised computer programs (e.g., public vs. private, encryption, buffer overflows, cybersecurity) (22)
- 2.5.8 Develop programs for multiple computing platforms (22)
- 2.5.9 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project (22)
- 2.5.10 Develop and use a series of test cases to verify that a program performs according to its design specifications (22)
- 2.5.11 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality) (22)

**CONTENT STANDARD 3.0: UNDERSTAND COMPUTING SYSTEMS****Performance Standard 3.1: Describe Devices**

- 3.1.1 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects (12)

**Performance Standard 3.2: Compare Hardware and Software**

- 3.2.1 Compare levels of abstraction and interactions between application software, system software, and hardware layers (12)
- 3.2.2 Categorize the roles of operating system software (12)
- 3.2.3 Discuss the use of computer virtualization (22)

**Performance Standard 3.3: Explain Troubleshooting**

- 3.3.1 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors (12)
- 3.3.2 Illustrate ways computing systems implement logic, input, and output through hardware components (12)

**CONTENT STANDARD 4.0: UNDERSTAND DATA AND ANALYSIS****Performance Standard 4.1: Evaluate Storage Solutions**

- 4.1.1 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) (12)
- 4.1.2 Demonstrate the ability to store bit representation of real-world phenomena, characters, numbers, and images (12)

**Performance Standard 4.2: Create Using Collection, Visualization, and Transformation**

- 4.2.1 Create interactive data visualizations or alternative representations using software tools to help others better understand real-world phenomena (12)
- 4.2.2 Use data analysis tools and techniques to identify patterns in data representing complex systems (12)
- 4.2.3 Select data collection tools and techniques to generate data sets that support a claim or communicate information (22)

---

**CONTENT STANDARD 5.0: UNDERSTAND IMPACTS OF COMPUTING****Performance Standard 5.1: Evaluate the Impact of Computing on Culture**

- 5.1.1 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices (12)
- 5.1.2 Test and refine computational artifacts to reduce bias and equity deficits (12)
- 5.1.3 Demonstrate ways a given algorithm applies to problems across disciplines (12)
- 5.1.4 Explain the potential impacts of artificial intelligence on society (12)
- 5.1.5 Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society (22)
- 5.1.6 Discuss how computational innovations that have revolutionized aspects of our culture might evolve (22)

**Performance Standard 5.2: Increase Social Interactions**

- 5.2.1 Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields (12)
- 5.2.2 Use tools and methods for collaboration to increase the productivity of a team (12)

**Performance Standard 5.3: Explain Safety, Law, and Ethics Related to Computing**

- 5.3.1 Explain the beneficial and harmful effects that intellectual property laws can have on innovation (12)
- 5.3.2 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users (12)
- 5.3.3 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics (12)
- 5.3.4 Discuss the role of ethics in technologies (22)
- 5.3.5 Discuss laws and regulations that impact the development and use of software (22)

**CONTENT STANDARD 6.0: UNDERSTAND NETWORKS AND THE INTERNET****Performance Standard 6.1: Evaluate Network, Communication, and Organization**

- 6.1.1 Evaluate the scalability and reliability of networks by describing the relationship between routers, switches, servers, topology, and addressing (22)
- 6.1.2 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology) (22)
- 6.1.3 Discuss the difference between deploying software to the cloud versus the standard model (i.e., running software locally on a PC) (22)

**Performance Standard 6.2: Describe Cybersecurity**

- 6.2.1 Illustrate how sensitive data can be affected by malware and other attacks (12)
- 6.2.2 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts (12)
- 6.2.3 Compare various security measures, considering tradeoffs between the usability and security of a computing system (12)
- 6.2.4 Explain tradeoffs when selecting and implementing cybersecurity recommendations (12)
- 6.2.5 Compare ways software developers protect devices and information from unauthorized access (e.g., password protection, 2 factor authentication) (22)
- 6.2.6 Utilize industry tools and best practices (e.g., code scanning tools, automated security scans during the development process) to create secure code that prevents vulnerabilities (22)

**CONTENT STANDARD 7.0: UNDERSTAND EMERGING TECHNOLOGIES**

**Performance Standard 7.1: Explain Workforce and Society Needs Related to New and Emerging Technologies**

- 7.1.1 Describe job skills needed for potential careers in new and emerging technologies (12)
- 7.1.2 Explore potential uses for and industries that may use emerging technologies (12)
- 7.1.3 Explain the role of ethics as it relates to security and emerging technologies (12)

## Complementary Courses

### State Complementary Skill Standards

State complementary skill standards are designed to clearly state what the student should know and be able to do upon completion of a **one-year** complementary course related to their career and technical education (CTE) program of study. **Completion of the qualifying Program of Study is required prior to enrollment in a complementary course.**

### Employability Skills for Career Readiness Standards

Students have completed all program content standards and will pursue advanced study through investigation and in-depth research.

### Complementary Course Standards Contributing Members

Course Contribution(s)	Name	Occupation/Title	Stakeholder Affiliation	School/Organization
Software and App Development	Ronald Barranco	Instructor	Secondary Educator	Southwest Career and Technical Academy, Clark County School District
Software and App Development	Frankie Clark	Instructor	Secondary Educator	North Valleys High School, Washoe County School District
Software and App Development	Josiah Davidson	Instructor	Secondary Educator	Advanced Technologies Academy, Clark County School District
Software and App Development	Roger Mayo	Instructor	Secondary Educator	Advanced Technologies Academy, Clark County School District
Software and App Development	Kathryn Milliken	Instructor	Secondary Educator	Damonte Ranch High School/Washoe County School District

### Business and Industry Validation

All CTE standards developed through the Nevada Department of Education are validated by business and industry through one or more of the following processes: (1) the standards are developed by a team consisting of business and industry representatives, or (2) a separate review panel is coordinated with industry experts to ensure the standards include the proper content, or (3) nationally recognized standards currently endorsed by business and industry.

The Software and App Development complementary standards for Advanced Computer Science program of study were validated through active participation of business and industry representatives through the criticality survey.

## Complementary Course Information for Advanced Computer Science

### Program Information

Qualifying Program of Study: **Advanced Computer Science**

Career Cluster: **Information Technology**

Career Pathway(s): **Programming and Software Development**

CTSO: **SkillsUSA / FBLA**

Grade Level: **11-12**

### Program Structure for Complementary Courses

The complementary courses are provided in the following table. **The qualifying program of study must be completed prior to enrolling in the complementary courses** (except labs that are done concurrently with the second-year course). A program does not have to utilize the complementary courses for students to complete their program of study.

#### Complementary Courses

Required/ Complementary	Course Title	Abbreviated Name
C	Software and App Development for Advanced Computer Science	APP DEV ADV COMP SCI
C	Advanced Computer Science Advanced Studies	ADV COMP SCI AS
C	Industry-Recognized Credential – Advanced Computer Science	IRC ADV COMP SCI
C	CTE Work Experience – Information Technology	WORK EXPER IT

---

## Complementary Course Standards Software and App Development

### CONTENT STANDARD 1.0: COLLABORATIVE DEVELOPMENT

#### Performance Standard 1.1: Apply Software Development Lifecycle (SDLC)

- 1.1.1 Describe the importance of the planning stage
- 1.1.2 Compare various development methodologies (e.g., Agile, Waterfall, etc.)
- 1.1.3 Apply appropriate methods for software/app development
- 1.1.4 Explain industry specific terminology (e.g., Scrum, sprint, Ansible, Docker, etc.)
- 1.1.5 Research the role of deployment tools and automation engines
- 1.1.6 Test code for performance and scalability
- 1.1.7 Check code for security flaws and vulnerabilities
- 1.1.8 Apply SDLC to manage projects

#### Performance Standard 1.2: Utilize Source Control Systems

- 1.2.1 Choose a source control system
- 1.2.2 Manage repositories
- 1.2.3 Explain branching and forking
- 1.2.4 Utilize push and pull requests
- 1.2.5 Handle merges and collisions
- 1.2.6 Implement major and minor version controls

### CONTENT STANDARD 2.0: TOOLS FOR DEVELOPMENT

#### Performance Standard 2.1: Evaluate Various Development Stacks

- 2.1.1 Select and plan the appropriate stack for frontend and backend of a project
- 2.1.2 Evaluate appropriate uses of various languages and databases
- 2.1.3 Explain the use of programming libraries
- 2.1.4 Explain the importance of dependencies
- 2.1.5 Apply appropriate data structures and corresponding algorithms

#### Performance Standard 2.2: Create Databases

- 2.2.1 Compare and evaluate different databases (e.g., Access, SQL, relational, no-SQL, etc.)
- 2.2.2 Design and implement databases
- 2.2.3 Perform basic database operations such as create, query, update, and delete records

#### Performance Standard 2.3: Utilize Application Programming Interface (API)

- 2.3.1 Describe the concept of API in software development
- 2.3.2 Explore and test popular API(s) used in industry
- 2.3.3 Implement an API in a software application
- 2.3.4 Create and document an API for an application

**Performance Standard 2.4: Integrated Development Environment (IDE)**

- 2.4.1 Explain the common features and uses of an IDE
- 2.4.2 Select and use an appropriate IDE
- 2.4.3 Explain the importance of software versions to an application
- 2.4.4 Use Software Development Kit (SDK) to develop and test applications
- 2.4.5 Research the impact of cloud computing on future IDE development

**CONTENT STANDARD 3.0: DELIVERABLES AND QUALITY CONTROL****Performance Standard 3.1: Create Documentation**

- 3.1.1 Create documentation for software and development projects
- 3.1.2 Use appropriate documentation tools and formats
- 3.1.3 Explain the importance of documentation in software and application development projects
- 3.1.4 Discuss the use of style guides in industry

**Performance Standard 3.2: Apply User Testing and Revision Practices**

- 3.2.1 Develop a testing plan for software and application development projects
- 3.2.2 Explain the role of dev vs. ops in application development workflows
- 3.2.3 Perform user testing
- 3.2.4 Revise software and app development projects based on user feedback
- 3.2.5 Document, track, and prioritize bugs and glitches

**Performance Standard 3.3 Describe the Importance of Deliverables**

- 3.3.1 Explain the rules and processes to publish an app through major distribution channels
- 3.3.2 Describe laws associated with publishing an app
- 3.3.3 Discuss the ethical implications of app development

**CONTENT STANDARD 4.0: TOPICS IN PROGRAMMING****Performance Standard 4.1: Develop and Maintain Portfolio**

- 4.1.1 Conduct peer and self-evaluations using rubrics
- 4.1.2 Explain the importance of maintaining a portfolio for software development
- 4.1.3 Identify strategies for ongoing learning and growth
- 4.1.4 Discuss professional networking with social media

**Performance Standard 4.2: Emerging Technologies in Computing**

- 4.2.1 Research careers and advancements in data science
- 4.2.2 Compare traditional app development to the vendor “store” model
- 4.2.3 Describe the role of cloud computing in app development
- 4.2.4 Research applications of artificial intelligence (AI) and machine learning in various industries
- 4.2.5 Research the role low-code and no-code platforms have on future application development