GRADES 9-12

Computer Science

IDAHO DEPARTMENT OF EDUCATION
CONTENT & CURRICULUM | TECHNOLOGY

650 W STATE STREET, 2ND FLOOR
BOISE, IDAHO 83702
208 332 6800 OFFICE / 711 TRS
WWW.SDE.IDAHO.GOV

CREATED 05/01/24

Contents

COMPUTING SYSTEMS

Code	Standard
9-12.CS.1.1	Identify and describe hardware components.
9-12.CS. 1.2	Identify and evaluate what computing system resources are required for a specific software program.
9-12.CS. 1.3	Identify the use of embedded computers in various applications.
9-12.CS. 1.4	Create or modify a program that uses different forms of input and output.
9-12.CS. 1.5	Identify how a high level programming language abstracts machine language in a computer program.
9-12.CS. 1.6	Create a model of how embedded systems sense, process, and interact in a given environment.

DATA AND ANALYSIS

Code	Standard
9-12.CS.2.1	Use applicable data collection techniques for various scenarios.
9-12.CS.2.2	Apply basic techniques for locating, collecting, and understanding the quality of data sets.
9-12.CS.2.3	Analyze data and identify patterns through modeling and simulation.
9-12.CS.2.4	Use data analysis to show the transformation from data to information to knowledge.
9-12.CS.2.5	Use models and simulations to help formulate, refine, and test scientific hypotheses.
9-12.CS.2.6	Compare and contrast the viewpoints on cybersecurity from the perspective of security experts, privacy advocates, and the government.

9-12.CS.2.7	Analyze the strengths and weaknesses of security policies based on their usage of encryption and authentication strategies.
9-12.CS.2.8	Convert between binary, decimal, octal, and hexadecimal representations of data.
9-12.CS.2.9	Describe how real-world phenomena such as numbers, Strings, or images are represented as binary in a computer.
9-12.CS.2.10	Analyze the trade-offs among various compression algorithms.

IMPACTS OF COMPUTING

Code	Standard
9-12.CS.3.1	Demonstrate responsible digital citizenship (legal and ethical behaviors) in the use of technology systems and software.
9-12.CS.3.2	Explain the social and economic implications associated with unethical computing practices.
9-12.CS.3.3	Discuss trade-offs such as privacy, safety, and convenience associated with the collection and large-scale analysis of personal information.
9-12.CS.3.4	Identify and evaluate the beneficial and harmful effects of computing innovations on behavior and culture.
9-12.CS.3.5	Debate how the issues of equity, data access, and distribution of computing resources create a digital divide in a global society.
9-12.CS.3.6	Debate laws and regulations that impact the development, security and use of software.
9-12.CS.3.7	Understand and define artificial intelligence.
9-12.CS.3.8	Research and explain the social, moral, ethical, and legal impacts of artificial intelligence systems and respective usage.

9-12.CS.3.9	Explain how computer automation continues to transform society and the global economy (e.g. financial markets, transactions, predictions).
9-12.CS.3.10	Research, analyze, and present how computational thinking has enabled computing to revolutionize business, manufacturing, commerce and society.
9-12.CS.3.11	Evaluate the accessibility of a computational artifact.
9-12.CS.3.12	Describe how computer science shares features with creating and designing an artifact such as in music and art.
9-12.CS.3.13	Understand the ecosystem of open--source software development and its impact on global collaboration.
9-12.CS.3.14	Explain how computer science fosters innovation and enhances other career and disciplines.

NETWORKS AND THE INTERNET

Code	Standard
9-12.CS.4.1	Illustrate the basic components of computer networks and protocols.
9-12.CS.4.2	Analyze the issues that impact network functionality.
9-12.CS.4.3	Describe the data flow that occurs when using Internet-based services.
9-12.CS.4.4	Examine how encryption is essential to ensuring privacy and security over the internet.

ALGORITHMS AND PROGRAMMING

Code	Standard
9-12.CS.5.1	Diagram the flow of execution and output of a given program.

9-12.CS.5.2	Design algorithms using sequence, selection, iteration and recursion.
9-12.CS.5.3	Use variable scope and encapsulation to design programs with cohesive and modular components.
9-12.CS.5.4	Decompose a complex problem using abstraction through methods and/or classes.
9-12.CS.5.5	Demonstrate the value of abstraction to manage problem complexity.
9-12.CS.5.6	Demonstrate code reuse by creating programming solutions using APIs and libraries.
9-12.CS.5.7	Evaluate the qualities of a program such as correctness, usability, readability, efficiency, portability and scalability through processes such as debugging and code review.
9-12.CS.5.8	Compare and contrast simple data structures and their uses.
9-12.CS.5.9	Compare software development processes.
9-12.CS.5.10	Demonstrate an understanding of the software life cycle process.
9-12.CS.5.11	Design and develop a software artifact by leading, initiating, and participating in a team.
9-12.CS.5.12	Create collaborative software projects using Integrated Development Environments, or other collaborative tools.
9-12.CS.5.13	Understand the positive and negative implications that arise when you add functionality to an existing program.
9-12.CS.5.14	Demonstrate how diverse team collaboration improves the design and development of software products.
9-12.CS.5.15	Compare a variety of programming languages available to solve problems and develop systems.
9-12.CS.5.16	Analyze security issues that might lead to compromised computer programs.
9-12.CS.5.17	Classify and define the different types of software licenses in order to understand how to apply each one to a specific software example.

9-12.CS.5.18	Analyze the notion of intelligent behavior through the programs that learn and adapt, play games, do image recognition, perform text analysis, and control the behavior of robots.
9-12.CS.5.19	Illustrate how mathematical and statistical functions, sets, and logic are used in computation.
9-12.CS.5.20	Describe the concept of parallel processing.
9-12.CS.5.21	Explore issues surrounding mobile computing.
9-12.CS.5.22	Explain the value of heuristic algorithms to approximate solutions for interactable problems.
9-12.CS.5.23	Critically examine algorithms and design an original algorithm (e.g. adapt, remix, improve).
9-12.CS.5.24	Classify problems as tractable, interactable, or computationally unsolvable.