Grades 9 – 12 Overview

Students in Grades 9-12 experience significant growth and development as they assume more complex responsibilities. They continue to develop unique personalities and begin to make important life decisions. In both school and community, high school students are strengthening and practicing the leadership and communication skills that facilitate entrance into adulthood. They continue to seek opportunities for realizing independence and individuality.

Grades 9-12 students have broadened their perspective regarding the importance of existing and developing technologies and have an understanding of the scope of technology in today's world. As students progress through their high school years, they are able to address a variety of problems on a range of topics in a logical manner. Technology offers students an efficient means for solving many types of problems.

Many students have opportunities to interact with people whose backgrounds are different from their own because of the cultural and ideological diversity of a technologically advanced global society. As the use of technology brings humankind closer together, concepts and skills utilizing digital literacy and computer science will assist students in becoming productive adults.

Grades 9-12 students will meet the following learning goals:

- As ***Computational Thinkers,*** students demonstrate how to simplify complex problems by developing algorithms that define the systematic processes.
- As ***Citizens of a Digital Culture,*** students demonstrate an understanding of concepts involving safety and security, responsible use of technology, and ways it can influence people through social interactions.
- As ***Global Collaborators,*** students utilize digital tools to collaborate and communicate with others to solve problems presented in today's technical world.
- As ***Computing Analysts,*** students analyze and create solutions to problems and challenges presented in the use of computer systems and data.
- As ***Innovative Designers,*** students make decisions and create solutions using the various digital tools available in today's technical environments.

Grades 9-12 Overview

Grades 9-12 content for digital literacy and computer science is organized into five strands of focused study outlined below in the column on the left and identified by bold print in shaded bars. Related content standards are grouped by topic below each strand.

The Recurring Standards for Digital Literacy and Computer Science are listed below in the column on the right. These recurring standards should be incorporated into classroom instruction at the appropriate level of rigor in each grade level.

Content Standard Strands and Topics

Computational Thinker

Abstraction
Algorithms
Programming and Development

Citizen of a Digital Culture

Safety, Privacy, and Security
Legal and Ethical Behavior
Digital Identity
Impact of Computing

Global Collaborator

Communication
Digital Tools
Collaborative Research

Computing Analyst

Data
Systems

Innovative Designer

Human/Computer Partnerships
Design Thinking

Recurring Standards

Safety, Privacy, and Security
1. Identify, demonstrate, and apply personal safe use of digital devices.

Legal and Ethical Behavior
2. Recognize and demonstrate age-appropriate responsible use of digital devices and resources as outlined in school/district rules.

Impact of Computing
3. Analyze the potential impact of computing.

Systems
4. Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.

Collaborative Research
5. Locate, curate, and evaluate information from digital sources to answer research questions.

Digital Tools
6. Produce, review, and revise authentic artifacts that include multimedia using appropriate digital tools.

Underlined words appear in the glossary.

Students can:

Computational Thinker

Abstraction

1. Decompose problems into component parts, extract key information, and develop descriptive models to understand the levels of abstractions in complex systems.
2. Explain how computing systems are often integrated with other systems and embedded in ways that may not be apparent to the user.
   Examples: Millions of lines of code control the subsystems within an automobile (e.g., antilock braking systems, lane detection, and self-parking).

Algorithms

3. Differentiate between a generalized expression of an algorithm in pseudocode and its concrete implementation in a programming language.
   a. Explain that some algorithms do not lead to exact solutions in a reasonable amount of time and thus approximations are acceptable.
   b. Compare and contrast the difference between specific control structures such as sequential statements, conditional, iteration, and explain the benefits and drawbacks of choices made.
      Examples: Tradeoffs involving implementation, readability, and program performance.
   c. Distinguish when a problem solution requires decisions to be made among alternatives, such as selection constructs, or when a solution needs to be iteratively processed to arrive at a result, such as iterative "loop" constructs or recursion.
   d. Evaluate and select algorithms based on performance, reusability, and ease of implementation.
   e. Explain how more than one algorithm may solve the same problem and yet be characterized with different priorities.
      Examples: All self-driving cars have a common goal of taking a passenger to a designation but may have different priorities such as safety, speed, or conservation; web search engines have their own algorithms for search with their own priorities.
4. Use and adapt classic algorithms to solve computational problems.
   Examples: Sorting, searching, shortest path, and data compression.

Programming and Development

5. Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using current events.
6. Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects, with parameters, and which return a result.
7. Compare and contrast fundamental data structures and their uses.
   Examples: Strings, lists, arrays, stacks, queues.
8. Demonstrate code reuse by creating programming solutions using libraries and Application Programming Interfaces.
9. Demonstrate the ability to verify the correctness of a program.
   a. Develop and use a series of test cases to verify that a program performs according to its design specifications

   b. Collaborate in a code review process to identify correctness, efficiency, scalability and readability of program code.

10. Resolve or debug errors encountered during testing using iterative design process.
    Examples: Test for infinite loops, check for bad input, check edge-cases.

Citizen of a Digital Culture

Safety, Privacy, and Security

11. Model and demonstrate behaviors that are safe, legal, and ethical while living, learning, and working in an interconnected digital world.
    a. Recognize user tracking methods and hazards.
       Examples: Cookies, WiFi packet sniffing.
    b. Understand how to apply techniques to mitigate effects of user tracking methods.
    c. Understand the ramifications of end-user license agreements and terms of service associated with granting rights to personal data and media to other entities.
    d. Explain the relationship between online privacy and personal security.
       Examples: Convenience and accessibility, data mining, digital marketing, online wallets, theft of personal information.
    e. Identify physical, legal, and ethical consequences of inappropriate digital behaviors.
       Examples: Cyberbullying/harassment, inappropriate sexual communications.
    f. Explain strategies to lessen the impact of negative digital behaviors and assess when to apply them.
12. Describe how sensitive data can be affected by malware and other attacks.
13. Compare various security measures of a computer system.
    Examples: Usability, security, portability, and scalability.
14. Compare ways to protect devices, software, and data.

Legal and Ethical Behavior

15. Explain the necessity for the school's Acceptable Use Policy.
16. Identify laws regarding the use of technology and their consequences and implications.
    Examples: Unmanned vehicles, net neutrality/common carriers, hacking, intellectual property, piracy, plagiarism.
17. Discuss the ethical ramifications of malicious hacking and its impact on society.
    Examples: Dissemination of privileged information, ransomware.
18. Explain the beneficial and harmful effects that intellectual property laws can have on innovation.

Digital Identity

19. Prove that digital identity is a reflection of persistent, publicly available artifacts.
20. Evaluate strategies to manage digital identity and reputation with awareness of the permanent impact of actions in a digital world.

Impact of Computing

21. Explain how technology facilitates the disruption of traditional institutions and services.
    Examples: Digital currencies, ridesharing, autonomous vehicles, retail, Internet of Things.
22. Research the impact of computing technology on possible career pathways.
    Examples: Government, business, medicine, entertainment, education, transportation.

23. Debate the positive and negative effects of computing innovations in personal, ethical, social, economic, and cultural spheres.
    Examples: Artificial Intelligence/machine learning, mobile applications, automation of traditional occupational skills.

Global Collaborator

Creative Communication

24. Compare and contrast Internet publishing platforms, including suitability for media types, target audience, and feedback mechanism.
    a. Apply version control capabilities within a digital tool to understand the importance of managing historical changes across suggestions made by a collaborative team.

Digital Tools

25. Utilize a variety of digital tools to create digital artifacts across content areas.

Collaborative Research

26. Use collaborative technologies to work with others including peers, experts, or community members to examine local, national, and global issues and problems from multiple viewpoints.

Social Interactions

27. Apply tools and methods for collaboration on a project to increase connectivity among people in different cultures and career fields.
    Examples: Collaborative documents, webinars, teleconferencing, and virtual fieldtrips

Computing Analyst

Data

28. Develop a model that reflects the methods, procedures and concepts used by computing devices in translating digital bits as real-world phenomena, such as print characters, sound, images, and video.
29. Summarize the role of compression and encryption in modifying the structure of digital artifacts and the varieties of information carried in the metadata of these artifacts.
30. Evaluate the tradeoffs involved in choosing methods for the organization of data elements and the location of data storage, including the advantages and disadvantages of networked computing.
    Examples: Client server, peer-to-peer, cloud computing.
31. Create interactive data visualizations using software tools to help others understand real-world phenomena.
32. Use data analysis tools and techniques to identify patterns in data representing complex systems.

Systems

33. Evaluate the scalability and reliability of networks by describing the relationship between routers, switches, servers, topology, packets, or addressing, as well as the issues that impact network functionality.
    Examples: Bandwidth, load, delay.
    a. Explain the purpose of Internet Protocol addresses and how domain names are resolved to IP addresses through a Domain Name System server.
    b. Understand the need for networking protocols and examples of common protocols. Examples: HTTP, SMTP, and FTP
34. Categorize the roles of operating system software.
35. Appraise the role of artificial intelligence in guiding software and physical systems. Examples: predictive modeling, self-driving cars.
36. Explain the tradeoffs when selecting and implementing cybersecurity recommendations. Examples: Two-factor authentication, password requirements, geolocation requirements.

Modeling and Simulation

37. Evaluate the ability of models and simulations to test and support the refinement of hypotheses.
    a. Create and utilize models and simulations to help formulate, test, and refine a hypothesis.
    b. Form a model of a hypothesis, testing the hypothesis by the collection and analysis of data generated by simulations. Examples: Science lab, robotics lab, manufacturing, space exploration.
    c. Explore situations where a flawed model provided an incorrect answer.

Innovative Designer

Human/Computer Partnerships

38. Systematically design and develop programs for broad audiences by incorporating feedback from users.
    Examples: Games, utilities, mobile applications.
39. Identify a problem that cannot be solved by either humans or machines alone and discuss a solution for it by decomposing the task into sub-problems suited for a human or machine to accomplish.
    Examples: Forecasting weather, piloting airplanes.

Design Thinking

40. Use an iterative design process, including learning from mistakes, to gain a better understanding of a problem domain.