

# Computer Science: Grades 9-12 Advanced

## Algorithms and Programming

- 1 Describe how artificial intelligence drives many software and physical systems.** [A9-12.AP.A.1](#)

---

- 2 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.** [A9-12.AP.A.2](#)

---

- 3 Use and adapt classic algorithms to solve computational problems.** [A9-12.AP.A.3](#)

---

- 4 Evaluate algorithms in terms of their efficiency, correctness, and clarity.** [A9-12.AP.A.4](#)

---

- 5 Compare and contrast fundamental data structures and their uses.** [A9-12.AP.V.1](#)

---

- 6 Illustrate the flow of execution of a recursive algorithm.** [A9-12.AP.C.1](#)

---

- 7 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.** [A9-12.AP.M.1](#)

---

- 8 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.** [A9-12.AP.M.2](#)

---

- 9 Demonstrate code reuse by creating programming solutions using libraries and APIs.** [A9-12.AP.M.3](#)

---

- 10 Plan and develop programs for broad audiences using a software life cycle process.** [A9-12.AP.PD.1](#)

---

- 11 Explain security issues that might lead to compromised computer programs.** [A9-12.AP.PD.2](#)

---

- 12 Develop programs for multiple computing platforms.** [A9-12.AP.PD.3](#)

---

- 13 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.** [A9-12.AP.PD.4](#)

---

- 14 Develop and use a series of test cases to verify that a program performs according to its design specifications.** [A9-12.AP.PD.5](#)

---

**15** Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality). [A9-12.AP.PD.6](#)

---

**16** Evaluate key qualities of a program through a process such as a code review. [A9-12.AP.PD.7](#)

---

**17** Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems. [A9-12.AP.PD.8](#)

---

## Computing Systems

**1** Categorize the roles of operating system software. [A9-12.CS.HS.1](#)

---

**2** Illustrate ways computing systems implement logic, input, and output through hardware components. [A9-12.CS.T.1](#)

---

## Data and Analysis

**1** Use data analysis tools and techniques to identify patterns in data representing complex systems. [A9-12.DA.CVT.1](#)

---

**2** Select data collection tools and techniques to generate data sets that support a claim or communicate information. [A9-12.DA.CVT.2](#)

---

**3** Evaluate the ability of models and simulations to test and support the refinement of hypotheses. [A9-12.DA.IM.1](#)

---

## Impacts of Computing

**1** Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society. [A9-12.IC.C.1](#)

---

**2** Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society. [A9-12.IC.C.2](#)

---

**3** Predict how computational innovations that have revolutionized aspects of our culture might evolve. [A9-12.IC.C.3](#)

---

**4** Debate laws and regulations that impact the development and use of software. [A9-12.IC.SLE.1](#)

---

## Networks and the Internet

**1** Describe the issues that impact network functionality (e.g., bandwidth, load, delay, and topology). [A9-12.NI.NCO.1](#)

---

**2** Compare ways software developers protect devices and information from unauthorized access. [A9-12.NI.C.1](#)

---