

Grades 9-12 L2

Computing Systems CS

D. Devices D

- 1 Describe how internal and external parts of computing devices function to form a system. L2.CS.D.01
-

HS. Hardware and Software HS

- 1 Categorize the roles of operating system software. L2.CS.HS.01
 - 2 Compare options for building a computer systems and determine advantages and drawbacks of each piece and how it will affect the overall performance. L2.CS.HS.02
-

IO. Input and Output IO

- 1 Demonstrate use of course specific advanced input and output devices related to field. L2.CS.IO.01
-

T. Troubleshooting T

- 1 Illustrate ways computing systems implement logic, input, and output through hardware components. L2.CS.T.01
-

Networks & the Internet NI

NCO. Network Communication & Organization NCO

- 1 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology). L2.NI.NCO.01
 - 2 Give examples to illustrate how sensitive data can be affected by malware and other attacks. L2.NI.NCO.02
-

C. Cybersecurity C

- 1 Compare ways software developers protect devices and information from unauthorized access. L2.NI.C.01
 - 2 Use encryption and decryption algorithms to transmit/ receive an encrypted message. L2.NI.C.02
-

Data Analysis DA

S. Storage S

- 1 Translate and compare different bit representations of data types, such as characters, numbers, and images. L2.DA.S.01
 - 2 Analyze file systems created for keeping track of files on the hard disk. L2.DA.S.02
 - 3 Select data collection tools and techniques to generate data sets that support a claim or communicate information. L2.DA.S.03
-

CVT. Visualization & Transformation CVT

- 1 Use data analysis tools and techniques to identify patterns in data representing complex systems. L2.DA.CVT.01
-

IM. Inference and Models IM

- 1 Evaluate the ability of models and simulations to test and support the refinement of hypotheses. (e.g., flocking behaviors, life cycles, etc.) L2.DA.IM.01
-

Algorithms and Programming AP

A. Algorithms A

- 1 Describe how artificial intelligence algorithms drive many software and physical systems (e.g., digital advertising, autonomous robots, computer vision, pattern recognition, text analysis). L2.AP.A.01
 - 2 Describe how artificial intelligence drives many software and physical systems. L2.AP.A.02
 - 3 Critically examine and trace classic algorithms (e.g., selection sort, insertion sort, binary search, linear search). L2.AP.A.03
 - 4 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem. L2.AP.A.04
 - 5 Use and adapt classic algorithms to solve computational problems. L2.AP.A.05
 - 6 Evaluate algorithms in terms of their efficiency, correctness, and clarity. L2.AP.A.06
-

V. Variables V

- 1 Compare and contrast simple data structures and their uses to simplify solutions, generalizing computational problems instead of repeatedly using primitive variables. L2.AP.V.01
-

C. Control C

- 1 Trace the execution of repetition (e.g., loops, recursion), illustrating output and changes in values of named variables. L2.AP.C.01

M. Modularity M

- 1 Construct solutions to problems using student-created components, such as procedures, modules and/or objects. L2.AP.M.01
- 2 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution. L2.AP.M.02
- 3 Demonstrate code reuse by creating programming solutions using libraries and APIs. L2.AP.M.03

PD. Program Development PD

- 1 Plan and develop programs for broad audiences using a software life cycle process. L2.AP.PD.01
- 2 Explain security issues that might lead to compromised computer programs. L2.AP.PD.02
- 3 Develop programs for multiple computing platforms. L2.AP.PD.03
- 4 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project. L2.AP.PD.04
- 5 Develop and use a series of test cases to verify that a program performs according to its design specifications. L2.AP.PD.05
- 6 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality). L2.AP.PD.06
- 7 Evaluate key qualities of a program through a process such as a code review. L2.AP.PD.07
- 8 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems. L2.AP.PD.08

Impacts of Computing IC

C. Culture C

- 1 Evaluate the beneficial and harmful effects that computational artifacts and innovations have on society. L2.IC.C.01
- 2 Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society. L2.IC.C.02
- 3 Design and implement a study that evaluates or predicts how computing has revolutionized an aspect of our culture and how it might evolve (e.g., education, healthcare, art/entertainment, energy). L2.IC.C.0

SI. Social Interactions SI

- 1 Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. L2.IC.SI.01

H. History H

- 1 Analyze trends of computing and how those trends have changed over time. L2.IC.H.01

SLE. Safety, Law, & Ethics SLE

- 1 Debate laws and regulations that impact the development and use of software. L2.IC.SLE.01
- 2 Determine ways to test the validity of information located online. L2.IC.SLE.02
- 3 Evaluate the social and economic consequences of how law and ethics interact with digital aspects of privacy, data, property, information, and identity. L2.IC.SLE.03

CP. Community Partnerships CP

- 1 Collaborate with local industry partners to design and implement a viable mentorship. L2.IC.CP.01