Missouri Computer Science

By the End of the 12th Grade

Computing Systems

Devices

1 Illustrate ways computing systems implement logic through hardware components. 11-12.CS.D.01

Hardware & Software

1 Describe and categorize roles of an operating system. 11-12.CS.HS.01

Troubleshooting

1 Describe how hardware components facilitate logic, input, output and storage in computing systems. 11-12.CS.T.01

Networks & the Internet

Network Communication & Organization

1 Analyze the relationship between routers, switches, servers, topology and addressing. 11-12.NI.NCO.01

2 Describe key protocols and underlying processes of internet-based services (e.g., http/https and Simple Mail Transfer Protocol (SMTP)/internet Message Access Protocol (IMAP), routing protocols). 11-12.NI.NCO.02

3 Explain how the characteristics of the internet influence the systems developed on it. 11-12.NI.NCO.03

Cybersecurity

1 Compare and refine ways in which software developers protect devices and information from unauthorized access. 11-12.NI.C.01

2 Analyze cryptographic techniques to model the secure transmission of information. 11-12.NI.C.02

Data & Analysis

Storage

1 Compare different bit representations of data types, such as characters, Booleans and numbers while recognizing when using each data type is appropriate. 11-12.DA.S.01

Collection, Visualization & Transformation

1 Generate data sets that use a variety of data collection tools and analysis techniques to support a claim and/or communicate information. 11-12.DA.CVT.01

Skip to main content

ıference & Models

1 Evaluate the ability of models and simulations to test and support the refinement of hypotheses. 11-12.DA.IM.01

Algorithms & Programming

Algorithms

1 Critically examine and trace classic algorithms (e.g., selection sort, insertion sort, binary search, linear search). 11-12.AP.A.01

2 Implement an artificial intelligence algorithm to interact with a human or solve a problem. 11-12.AP.A.02

3 Describe how artificial intelligence algorithms drive many software and physical systems (e.g., autonomous robots, computer vision, pattern recognition, text analysis). 11-12.AP.A.03

4 Evaluate algorithms (e.g., sorting, searching) in terms of their efficiency and clarity. 11-12.AP.A.04

Variables

1 Create problem solutions that utilize data structures (e.g., lists, arrays, ArrayLists). 11-12.AP.V.01

Control

1 Trace the execution of iteration (e.g., loops, recursion), illustrating output and changes in values of named variables. 11-12.AP.C.01

Modularity

1 Construct solutions to problems using student- created components (e.g., procedures, modules, objects). 11-12.AP.M.01

2 Create programming solutions by reusing existing code (e.g., libraries, Application Programming Interface (APIs), code repositories). 11-12.AP.M.02

3 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution. 11-12.AP.M.03

Program Development

1 Use integrated development environments (IDEs) and collaborative tools and practices (code documentation) in a software project. 11-12.AP.PD.01

2 Plan and develop programs using a development process (e.g., waterfall, iterative, spiral, rapid application development, agile). 11-12.AP.PD.02

3 Identify and compare features of various programming languages that make them useful for solving problems and developing systems. 11-12.AP.PD.03

4 Design software using version control. 11-12.AP.PD.04

5 Develop and use a series of test cases to verify that a program performs according to its design specifications. 11-12.AP.PD.05

Explain security issues that might lead to compromised computer programs. 11-12.AP.PD.06

Skip to main content

7 Evaluate key qualities of a program through a process such as a code review. 11-12.AP.PD.07

Impacts of Computing

Culture

1 Evaluate the impact of equity, access and influence on the distribution of computing resources in a global society. 11-12.IC.C.01

Safety, Law & Ethics

1 Debate laws and regulations that impact the development and use of software. 11-12.IC.SLE.01