

Introduction to CS

Computing Systems

Devices

- 1 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. [ICS-CS-01](#)
-

Hardware & Software

- 2 Compare levels of abstraction and interactions between application software, system software, and hardware layers. [ICS-CS-02](#)
 - 3 Explain the roles of operating systems including memory management, data storage/retrieval, process management, and access control. [ICS-CS-03](#)
-

Troubleshooting

- 4 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. [ICS-CS-04](#)
-

Networks & The Internet

Network Communication & Organization

- 1 Evaluate the relationship between routers, switches, servers, and topology with regard to networks. [ICS-NI-01](#)
-

Cybersecurity

- 2 Identify examples to illustrate how sensitive data can be affected by malware and other attacks. [ICS-NI-02](#)
 - 3 Recommend cybersecurity measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. [ICS-NI-03](#)
 - 4 Compare various security measures and consider tradeoffs between the usability and security of a computing system. [ICS-NI-04](#)
-

Data & Analysis

Storage

- 1 Compare different binary representations of data, including text, sound, images, and numbers. [ICS-DA-01](#)
 - 2 Evaluate the tradeoffs in how data elements are organized and where data is stored. [ICS-DA-02](#)
-

Collection, Visualization, & Transformation

- 3 Create interactive data visualizations using software tools to help others better understand real-world phenomena. [ICS-DA-03](#)

Inference & Models

- 4 Create computational models that represent the relationships among different elements of data collected. [ICS-DA-04](#)

Algorithms & Programming

Algorithms

- 1 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. [ICS-AP-01](#)
- 2 Explain the use of artificial intelligence within computing systems. [ICS-AP-02](#)

Variables

- 3 Utilize lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. [ICS-AP-03](#)

Control

- 4 Justify the selection of specific control structures, considering implementation, readability, and program performance. [ICS-AP-04](#)
- 5 Iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. [ICS-AP-05](#)

Modularity

- 6 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. [ICS-AP-06](#)

Program Development

- 7 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. [ICS-AP-07](#)
 - 8 Systematically design programs for broad audiences. [ICS-AP-08](#)
 - 9 Refine programs by incorporating feedback from users. [ICS-AP-09](#)
 - 10 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as software libraries. [ICS-AP-10](#)
 - 11 Evaluate computational artifacts for usability. [ICS-AP-11](#)
 - 12 Modify computational artifacts to increase usability and accessibility. [ICS-AP-12](#)
 - 13 Develop computational artifacts working in team roles using collaborative tools. [ICS-AP-13](#)
 - 14 Explain design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. [ICS-AP-14](#)
-

Impacts of Computing

Culture

- 1 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. [ICS-IC-01](#)
 - 2 Elaborate how computational innovations have and may continue to impact society. [ICS-IC-02](#)
 - 3 Evaluate how equity, access, and influence impact distribution of computing resources in a global society. [ICS-IC-03](#)
 - 4 Test computational artifacts to reduce bias and equity deficits. [ICS-IC-04](#)
 - 5 Demonstrate ways a given algorithm applies to problems across disciplines. [ICS-IC-05](#)
-

Social Interactions

- 6 Utilize tools and methods for collaboration on a project to increase connectivity of peers. [ICS-IC-06](#)
-

Safety, Law, & Ethics

- 7 Explain the beneficial and harmful effects that intellectual property laws can have on innovation. [ICS-IC-07](#)
- 8 Explain privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. [ICS-IC-08](#)
- 9 Evaluate the social and economic implications of privacy in the context of safety, law, and ethics. [ICS-IC-09](#)
- 10 Explain how laws and regulations impact the development and use of software. [ICS-IC-10](#)