Competency Consolidation

# Table of Contents

[Table of Contents](#_m8mg73rotfe4)

[Participants](#_wmz6kom7eysm)

[Overview](#_k4ml8an2knlt)

[Current Big Ideas/Competencies](#_7mnkf5z9acv9)

[Computer Systems Networking](#_77jhej9ghap2)

[Computer Programming](#_5yyxrwvnou7u)

[Computer Repair](#_m3jyi84dpebs)

[Pre-Existing Common Competencies](#_a74hy15r6bou)

[Existing Frameworks for Computer Science](#_ksz6qhqy3fl1)

[K–12 Computer Science Framework](#_jink5gask6w1)

[Core Concepts](#_6c1e4s9k1v7a)

[Core Practices](#_ygxnj3igg9kv)

[California K–12 Computer Science Standards](#_l1jpnwbzcgsx)

[Colorado CS Academic Standards](#_dah1n5s03vkp)

[NH CS Standards](#_92et2off7l5a)

[Career Pathways associated with Content](#_m7cklxxdcm3v)

[Proposed Competency Consolidation](#_t9ymajn7bi33)

[Computer Programming Condensed Competency Ideas](#_l767rh4g6b92)

[Algorithms and Programming](#_35q9az6qsfv5)

[Networking Condensed Competency Ideas](#_owdyp0lvsp2g)

[Networks and the Internet](#_rgbtimmwrz6w)

[Computer Hardware Condensed Competency Ideas](#_nnopyrq6wa4)

[Computing Hardware](#_q4p98xdi2j9e)

[Additional Competency Proposal](#_uz485xz00rsb)

[Data and Analysis](#_o47yaoah7ls)

[Career Ready](#_rlwwn3n9ukk1)

[New Competencies](#_kud5oqduxa7r)

[New Competency Statements](#_wm4onsr8dwdy)

[Competency Breakdown](#_26nh9o21ij1r)

[Algorithms and Programming](#_69dvj9c1lvuf)

[Competency Breakdown:](#_q2v7fb8xc5cw)

[Networks and the Internet](#_jinposfheezv)

[Competency Breakdown:](#_e3nq2i6cbtau)

[Computing Hardware](#_3tjkldea1l)

[Competency Breakdown:](#_8rp6t8uqhy5r)

[Career Ready](#_sn28y0bl3mdl)

[Competency Breakdown:](#_yshdcktgjkhd)

[Data Analysis](#_vh1ire52iqft)

[Example Class Structure](#_mwmlms9td1pk)

[Physical Programming](#_ctn0078mlda7)

[Cybersecurity](#_y1gn1j2sh3c)

[AP Computer Science A](#_z4rx9j7cp8xi)

### Participants

Mike Eno

# 

# Overview

//description of task for group

//Name and CIP of new CIP code: 11.0101: Computer and Information Sciences, General

# Current Big Ideas/Competencies

## Computer Systems Networking

* Understand the basic concepts of PC computer assembly, troubleshooting and repair in order to effectively troubleshoot and repair equipment in a timely manner.
* Understand the basic concepts of network construction in order to design, build and connect hardware to allow people to communicate securely in a timely fashion.
* Understand the basic concepts of operating a network infrastructure in order to select the proper network protocols to build a secure network.
* Understand communication concepts and techniques in order to provide quality customer/client service

## Computer Programming

* Understand the basic principles of computer program development in order to create a foundation on which to base more complex software design.
* Understand the fundamentals of programming languages that are critical to the creation of methods and the concept of structured programming
* Understand event handling and user interaction in order to understand data flow and control.
* Understand the basic common algorithms of computer science to show how they affect ways to solve mathematical or programming problems.
* Understand the basic concepts of computer networks and security to reinforce knowledge of ethical computing.

## Computer Repair

* Understand proper use of tools and equipment, and preventative maintenance practices to provide a safe work environment
* Understand and utilize fundamental electrical concepts in order to insure safe connections.
* Comprehend basic hardware principles, methods, and procedures in order to set up and troubleshoot systems
* Comprehend basic system hardware principles, methods, and procedures in order to set up and troubleshoot systems
* Understand software application concepts and utilization in order to enhance customer service
* Comprehend and utilize computer operating systems in order to repair and maintain a system
* Comprehend basic maintenance principles, methods, and procedures to prevent system failure
* Understand basic troubleshooting methods, procedures, and techniques to troubleshoot and repair a system
* Understand customer service concepts and practices to effectively communicate with customers
* Comprehend methods used to locate and utilize support resources in order to find solutions to problems
* Understand basic network concepts to interconnect systems
* Comprehend network principles and procedures to repair and maintain the networked system

## Pre-Existing Common Competencies

These are competencies which exist in all three current zip codes.

* Understand the fundamental concepts of entrepreneurship and how entrepreneurship influences the economy
* Understand the importance of personal growth and leadership to enhance career success
* Understand the necessary employability Skill Identifiers in order to achieve success in today’s workplace

# Existing Frameworks for Computer Science

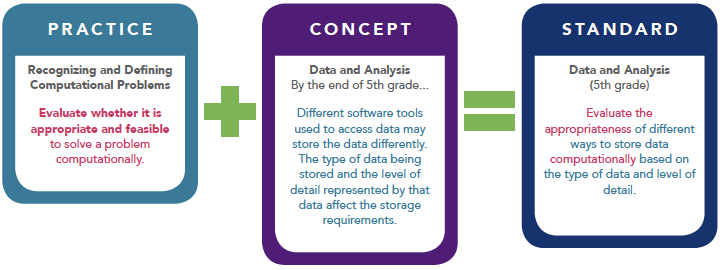
## [K–12 Computer Science Framework](https://k12cs.org/)

### Core Concepts

* Computing Systems
* Networks and the Internet
* Data and Analysis
* Algorithms and Programming
* Impacts of Computing

### Core Practices

* Fostering an Inclusive Computing Culture
* Collaborating Around Computing
* Recognizing and Defining Computational Problems
* Developing and Using Abstractions
* Creating Computational Artifacts
* Testing and Refining Computational Artifacts
* Communicating About Computing



## [California K–12 Computer Science Standards](https://drive.google.com/file/d/11iaS9JUZMtp0Qxqa4VQeot-IUohuDed_/view?usp=sharing)

## [Colorado CS Academic Standards](https://www.cde.state.co.us/computerscience/2020cas-cs-912)

## [NH CS Standards](https://www.education.nh.gov/sites/g/files/ehbemt326/files/inline-documents/standards-cs1.pdf)

# 

# Career Pathways associated with Content

* Web Developer
* Mobile Application Developer
* Software Engineer
* Network Administrator
* IT Help Desk Technician
* IT Manager
* Database Administrator
* IT Security Specialist
* Systems Analyst
* Computer Programmer
* IT Project Manager
* Computer Scientist
* Computer Hardware Engineer
* Robotics Engineer
* Network Architect
* DevOps Engineer
* Data Scientist
* User Interface Developer

# 

# 

# Proposed Competency Consolidation

## Computer Programming Condensed Competency Ideas

This is the competency that applies to programming and would lead to careers like Web Developer, Mobile application Developer, Software Engineer, computer programmer, etc. These combine some of the current competencies in one overall one focusing on design of code and algorithms.

### Algorithms and Programming

*An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are translated into programs, or code, to provide instructions for computing devices. Algorithms and programming control all computing systems, empowering people to communicate with the world in new ways and solve compelling problems. The development process to create meaningful and efficient programs involves choosing which information to use and how to process and store it, breaking apart large problems into smaller ones, recombining existing solutions, and analyzing different solutions.*

| **Skill Identifier** | **Concept** | **Description** |
| --- | --- | --- |
| AP-01 | Program Design | Diverse teams can develop programs with a broad impact through careful review and by drawing on the strengths of members in different roles. Design decisions often involve trade offs. The development of complex programs is aided by resources such as libraries and tools to edit and manage parts of the program. Systematic analysis is critical for identifying the effects of lingering bugs. |
| AP-02 | Algorithms | People evaluate and select algorithms based on performance, reusability, and ease of implementation. Knowledge of common algorithms improves how people develop software, secure data, and store information. |
| AP-03 | Variables | Data structures are used to manage program complexity. Programmers choose data structures based on functionality, storage, and performance tradeoffs. |
| AP-04 | Control | Programmers consider tradeoffs related to implementation, readability, and program performance when selecting and combining control structures. |
| AP-05 | Modularity | Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. These modules can be procedures within a program; combinations of data and procedures; or independent, but interrelated, programs. Modules allow for better management of complex tasks. |

## Networking Condensed Competency Ideas

This is the competency that applies to networking. This is everything from the basics of the internet, OSI model, creating peer to peer networks, etc. This is for those looking for career paths like Network architect, IT specialist, Help Desk, Network admin, etc.

### Networks and the Internet

*Computing devices typically do not operate in isolation. Networks connect computing devices to share information and resources and are an increasingly integral part of computing. Networks and communication systems provide greater connectivity in the computing world by providing fast, secure communication and facilitating innovation.*

| **Skill Identifier** | **Concept** | **Description** |
| --- | --- | --- |
| NI-01 | Network Construction | Network topology is determined, in part, by how many devices can be supported. Each device is assigned an address that uniquely identifies it on the network. The scalability and reliability of the Internet are enabled by the hierarchy and redundancy in networks. |
| NI-02 | Cybersecurity | Network security depends on a combination of hardware, software, and practices that control access to data and systems. The needs of users and the sensitivity of data determine the level of security implemented. |
| NI-03 | System Administration | A system administrator, or sysadmin, is a person who is responsible for the upkeep, configuration, and reliable operation of computer systems; especially multi-user computers, such as servers. |

## 

## Computer Hardware Condensed Competency Ideas

This is the competency that applies to Hardware and identifying parts of a computer and other devices.

### Computing Hardware

*People interact with a wide variety of computing devices that collect, store, analyze, and act upon information in ways that can affect human capabilities both positively and negatively. The physical components (hardware) and instructions (software) that make up a computing system communicate and process information in digital form. An understanding of hardware and software is useful when troubleshooting a computing system that does not work as intended.*

| **Skill Identifier** | **Concept** | **Description** |
| --- | --- | --- |
| CS-01 | Safety | Practice proper electrical safety. Demonstrate proper tool practices and techniques. Apply proper electrostatic discharge procedures. Differentiate between low and high voltage AC and DC electrical systems. Demonstrate proper care of materials, tools, and equipment. Describe the four basic electrical quantities. Describe the relationship between electricity and magnetism. Measure current, voltage and resistance. |
| CS-02 | Devices | Computing devices are often integrated with other systems, including biological, mechanical, and social systems. These devices can share data with one another. The usability, dependability, security, and accessibility of these devices, and the systems they are integrated with, are important considerations in their design as they evolve. |
| CS-03 | Hardware and Software | Levels of interaction exist between the hardware, software, and user of a computing system. The most common levels of software that a user interacts with include system software and applications. System software controls the flow of information between hardware components used for input, output, storage, and processing. |
| CS-04 | Troubleshooting | Troubleshooting complex problems involves the use of multiple sources when researching, evaluating, and implementing potential solutions. Troubleshooting also relies on experience, such as when people recognize that a problem is similar to one they have seen before or adapt solutions that have worked in the past. |

## Additional Competency Proposal

This is for collecting data and using it as a visual aid. This is for careers in Data Science, Computer Scientist, Data Analyst, Database manager, etc.

### Data and Analysis

*Computing systems exist to process data. The amount of digital data generated in the world is rapidly expanding, so the need to process data effectively is increasingly important. Data is collected and stored so that it can be analyzed to better understand the world and make more accurate predictions.*

| **Skill Identifier** | **Concept** | **Description** |
| --- | --- | --- |
| DA-01 | Collection | Data is constantly collected or generated through automated processes that are not always evident, raising privacy concerns. The different collection methods and tools that are used influence the amount and quality of the data that is observed and recorded. |
| DA-02 | Storage | Data can be composed of multiple data elements that relate to one another. For example, population data may contain information about age, gender, and height. People make choices about how data elements are organized and where data is stored. These choices affect cost, speed, reliability, accessibility, privacy, and integrity. |
| DA-03 | Visualization and Transformation | People transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information. Examples include visualization, aggregation, rearrangement, and application of mathematical operations. |
| DA-04 | Inference and Models | The accuracy of predictions or inferences depends upon the limitations of the computer model and the data the model is built upon. The amount, quality, and diversity of data and the features chosen can affect the quality of a model and ability to understand a system. Predictions or inferences are tested to validate models. |

This is an overall catch all for career readiness in a specialized field of technology

### Career Ready

*Computing affects many aspects of the world in both positive and negative ways at local, national, and global levels. Individuals and communities influence computing through their behaviors and cultural and social interactions, and in turn, computing influences new cultural practices. An informed and responsible person should understand the social implications of the digital world, including equity and access to computing. How do we prepare for a career in technology?*

| **Skill Identifier** | **Concept** | **Description** |
| --- | --- | --- |
| CR-01 | Culture | The design and use of computing technologies and artifacts can improve, worsen, or maintain inequitable access to information and opportunities. |
| CR-02 | Social Interactions | Many aspects of society, especially careers, have been affected by the degree of communication afforded by computing. The increased connectivity between people in different cultures and in different career fields has changed the nature and content of many careers. |
| CR-03 | Safety, Law, and Ethics | Laws govern many aspects of computing, such as privacy, data, property, information, and identity. These laws can have beneficial and harmful effects, such as expediting or delaying advancements in computing and protecting or infringing upon people’s rights. International differences in laws and ethics have implications for computing. |
| CR-04 | Career Plan / Portfolio | Fundamental concepts of entrepreneurship and how entrepreneurship influences the economy. The importance of personal growth and leadership to enhance career success. The necessary employability Skill Identifiers in order to achieve success in today’s workplace |

# New Competencies

This ideally is what we would report to the state. Each one will have sub categories but we take the average of the sub category at the end of a reporting period and submit a value of 1 - 4 for each big idea category

| **Competency Name/Big Idea** | **Description** |
| --- | --- |
| **Algorithms and Programming** | An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are translated into programs, or code, to provide instructions for computing devices. Algorithms and programming control all computing systems, empowering people to communicate with the world in new ways and solve compelling problems. The development process to create meaningful and efficient programs involves choosing which information to use and how to process and store it, breaking apart large problems into smaller ones, recombining existing solutions, and analyzing different solutions. |
| **Networks and the Internet** | Computing devices typically do not operate in isolation. Networks connect computing devices to share information and resources and are an increasingly integral part of computing. Networks and communication systems provide greater connectivity in the computing world by providing fast, secure communication and facilitating innovation. |
| **Computing Systems** | People interact with a wide variety of computing devices that collect, store, analyze, and act upon information in ways that can affect human capabilities both positively and negatively. The physical components (hardware) and instructions (software) that make up a computing system communicate and process information in digital form. An understanding of hardware and software is useful when troubleshooting a computing system that does not work as intended. |
| **Data and Analysis** | Computing systems exist to process data. The amount of digital data generated in the world is rapidly expanding, so the need to process data effectively is increasingly important. Data is collected and stored so that it can be analyzed to better understand the world and make more accurate predictions. |
| **Cybersecurity** | Prove how to detect, prevent and mitigate threats in order to secure a computing system or network in an ethical manner, and in accordance with international, federal,state, local and cyber laws and regulations |

# 

# New Competency Statements

| **Competency Name/Big Idea** | **Statement** |
| --- | --- |
| **Algorithms and Programming** | Students will expand their use of software design process in the application of algorithm and program design in order to make use of the principles of programming. |
| **Networks and the Internet** | Students will... |
| **Computing Systems** | Students will... |
| **Data and Analysis** | Students will expand their understanding on how computing systems exist to process data, simulate data and evaluate data in a visual way. |
| **Cybersecurity** | Students will... |

# 

# 

| Algorithms and Programming *An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are translated into programs, or code, to provide instructions for computing devices. Algorithms and programming control all computing systems, empowering people to communicate with the world in new ways and solve compelling problems. The development process to create meaningful and efficient programs involves choosing which information to use and how to process and store it, breaking apart large problems into smaller ones, recombining existing solutions, and analyzing different solutions.* | | | | |
| --- | --- | --- | --- | --- |
| **Skill Identifier** | **Level 1** | **Level 2** | **Level 3** | **Level 4** |
| **AP-01** | Students develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems. | Students develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. Students continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs. | Students design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met. | Diverse Student teams can develop programs with a broad impact through careful review and by drawing on the strengths of members in different roles. Design decisions often involve trade offs. The development of complex programs is aided by resources such as libraries and tools to edit and manage parts of the program. Systematic analysis is critical for identifying the effects of lingering bugs. |
| **AP-02** | Students follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow. | Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. | Algorithms affect how people interact with computers and the way computers respond. Students design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug. | Students evaluate and select algorithms based on performance, reusability, and ease of implementation. Knowledge of common algorithms improves how people develop software, secure data, and store information. |
| **AP-03** | Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it. | Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. | Programmers create variables to store data values of selected types. A meaningful Skill Identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs. | Data structures are used to manage program complexity. Programmers choose data structures based on functionality, storage, and performance tradeoffs. |
| **AP-04** | Computers follow precise sequences of instructions that automate tasks. Program execution can also be non sequential by repeating patterns of instructions and using events to initiate instructions. | Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. | Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior. | Programmers consider tradeoffs related to implementation, readability, and program performance when selecting and combining control structures. |
| **AP-05** | Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks. | Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. | Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability. | Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. These modules can be procedures within a program; combinations of data and procedures; or independent, but interrelated, programs. |

| **Skill Identifier** | **Concept** | **Level 1** |
| --- | --- | --- |
| AP-01 | **Program Design** | * Identify and use an iterative process to plan the development of a program by including others' perspectives and considering user preferences. * Define Graphical User Interfaces common to apps * State an app’s technical documentation content and purpose. * Identify Debugging tools and processes. * Observe intellectual property rights and give appropriate attribution when creating or remixing programs. * Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. * Describe choices made during program development using code comments, presentations, and demonstrations. |
| AP-02 | **Algorithms** | * Compare and refine multiple algorithms for the same task and determine which is the most appropriate. * Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. * Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. |
| AP-03 | **Variables** | * Create programs that use variables to store and modify data. Variables are used to store and modify data. * Identify Data types and memory models |
| AP-04 | **Control** | * Create programs that include sequences, events, loops, and conditionals. |
| AP-05 | **Modularity** | * Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. |

| **Skill Identifier** | **Concept** | **Level 2** |
| --- | --- | --- |
| AP-01 | **Program Design** | * Seek and incorporate feedback from team members and users to refine a solution that meets user needs * Incorporate existing code, media, and libraries into original programs, and give attribution. * Systematically test and refine programs using a range of test cases. * Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. * Document programs in order to make them easier to follow, test, and debug. * Explain the process of agile and waterfall development process and wireframing. |
| AP-02 | **Algorithms** | * Use flowcharts and/or pseudocode to address complex problems as algorithms. |
| AP-03 | **Variables** | * Create clearly named variables that represent different data types and perform operations on their values. |
| AP-04 | **Control** | * Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. |
| AP-05 | **Modularity** | * Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. * Create procedures with parameters to organize code and make it easier to reuse. |

| **Skill Identifier** | **Concept** | **Level 3** |
| --- | --- | --- |
| AP-01 | **Program Design** | * Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. * Systematically design and develop programs for broad audiences by incorporating feedback from users. * Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. * Evaluate and refine computational artifacts to make them more usable and accessible * Design and develop computational artifacts working in team roles using collaborative tools * Document –design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. |
| AP-02 | **Algorithms** | * Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. |
| AP-03 | **Variables** | * Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. |
| AP-04 | **Control** | * Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. |
| AP-05 | **Modularity** | * Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. * Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. |

| **Skill Identifier** | **Concept** | **Level 4** |
| --- | --- | --- |
| AP-01 | **Program Design** | * Describe how artificial intelligence drives many software and physical systems * Plan and develop programs for broad audiences using a software lifecycle process * Explain security issues that might lead to compromised computer programs * Develop programs for multiple computing platforms * Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project * Develop and use a series of test cases to verify that a program performs according to its design specifications * Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality). * Evaluate key qualities of a program through a process such as a code review. * Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems. |
| AP-02 | **Algorithms** | * Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem. * Use and adapt classic algorithms to solve computational problems. * Evaluate algorithms in terms of their efficiency, correctness, and clarity. |
| AP-03 | **Variables** | * Compare and contrast fundamental data structures and their uses. |
| AP-04 | **Control** | * Illustrate the flow of execution of a recursive algorithm. |
| AP-05 | **Modularity** | * Construct solutions to problems using student-created components, such as procedures, modules and/or objects. * Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution. * Demonstrate code reuse by creating programming solutions using libraries and APIs |

# 

# 

| Networks and the Internet *Computing devices typically do not operate in isolation. Networks connect computing devices to share information and resources and are an increasingly integral part of computing. Networks and communication systems provide greater connectivity in the computing world by providing fast, secure communication and facilitating innovation.* | | | | |
| --- | --- | --- | --- | --- |
| **Skill Identifier** | **Level 1** | **Level 2** | **Level 3** | **Level 4** |
| **NI-01** | Computer networks can be used to connect people to other people, places, information, and ideas. The Internet enables people to connect with others worldwide through many different points of connection. | Information needs a physical or wireless path to travel to be sent and received, and some paths are better than others. Information is broken into smaller pieces, called packets, that are sent independently and reassembled at the destination. Routers and switches are used to properly send packets across paths to their destinations. | Computers send and receive information based on a set of rules called protocols. Protocols define how messages between computers are structured and sent. Considerations of security, speed, and reliability are used to determine the best path to send and receive data. | Network topology is determined, in part, by how many devices can be supported. Each device is assigned an address that uniquely identifies it on the network. The scalability and reliability of the Internet are enabled by the hierarchy and redundancy in networks. |
| **NI-02** | Connecting devices to a network or the Internet provides great benefit, but care must be taken to use authentication measures, such as strong passwords, to protect devices and information from unauthorized access. | Information can be protected using various security measures. These measures can be physical and/or digital. | The information sent and received across networks can be protected from unauthorized access and modification in a variety of ways, such as encryption to maintain its confidentiality and restricted access to maintain its integrity. Security measures to safeguard online information proactively address the threat of breaches to personal and private data. | Network security depends on a combination of hardware, software, and practices that control access to data and systems. The needs of users and the sensitivity of data determine the level of security implemented. |
| **NI-03** |  |  | A system administrator, or sysadmin, is a person who is responsible for the upkeep, configuration, and reliable operation of computer systems; especially multi-user computers, such as servers. |  |

| **Skill Identifier** | **Concept** | **Level 1** |
| --- | --- | --- |
| NI-01 | **Network Construction** | * Identify basic terms, concepts, and functions of system modules, including how each module works during normal operation. * Identify basic procedures for adding and removing field replaceable modules. * Identify the necessary system resources and procedures for configuring them for device installation. * Identify common peripheral ports, associated cabling, and their connectors. |
| NI-02 | **Cybersecurity** | * Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access. * Discuss real-world cybersecurity problems and how personal information can be protected. |
| NI-03 | **System Administration** | * Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination. * Identify the basic procedure for installing OS software, including DOS and Windows. * Identify the purpose of BIOS, what it contains, and how to change it. * Identify the basic functions, components and interfaces of the major commercial and open source operating systems. * Identify the proper procedure for removing and creating partitions on a hard drive. * Identify the basic system boot sequences and boot methods, including the steps to remedy Boot problems. |

| **Skill Identifier** | **Concept** | **Level 2** |
| --- | --- | --- |
| NI-01 | **Network Construction** | * Demonstrate a basic knowledge of topologies, segments, and backbones. * Explain the layers of the OSI and TCP/IP Models. * Describe the major network operating systems and their similarities and differences. * Explain the characteristics of networking standards. * Describe and contrast LAN and WAN technologies and virtual networks. * Explain network planning and documentation |
| NI-02 | **Cybersecurity** | * Explain how physical and digital security measures protect electronic information. * Apply multiple methods of encryption to model the secure transmission of information. * Explain the characteristics of various types of malware and the means of prevention. |
| NI-03 | **System Administration** | * Model the role of protocols in transmitting data across networks and the Internet * Recognize and describe the advantages and disadvantages of various networking media. * Describe the major advantages and disadvantages of share folder and file level security on a network. |

| **Skill Identifier** | **Concept** | **Level 3** |
| --- | --- | --- |
| NI-01 | **Network Construction** | * Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. * Use TCP/IP troubleshooting utilities such as PING and Tracert/Traceroute to test IP connectivity. * Select, Terminate and test media |
| NI-02 | **Cybersecurity** | * Give examples to illustrate how sensitive data can be affected by malware and other attacks * Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. * Compare various security measures, considering tradeoffs between the usability and security of a computing system. * Explain trade offs when selecting and implementing cybersecurity recommendations. |
| NI-03 | **System Administration** | * Design a peer-to-peer network. * Design a client-server network. * Create and administer groups, users, application and data folders and allocate user system resources. * Install File and folder level security on a network. * Install and share a printing device on a network. |

| **Skill Identifier** | **Concept** | **Level 4** |
| --- | --- | --- |
| NI-01 | **Network Construction** | * Create a peer-to-peer network. * Create a client-server network. |
| NI-02 | **Cybersecurity** | * Compare ways software developers protect devices and information from unauthorized access |
| NI-03 | **System Administration** | * Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology). |

# 

# 

| System Administration (Hardware) *People interact with a wide variety of computing devices that collect, store, analyze, and act upon information in ways that can affect human capabilities both positively and negatively. The physical components (hardware) and instructions (software) that make up a computing system communicate and process information in digital form. An understanding of hardware and software is useful when troubleshooting a computing system that does not work as intended.* | | | | |
| --- | --- | --- | --- | --- |
| **Skill Identifier** | **Level 1** | **Level 2** | **Level 3** | **Level 4** |
| **SAH-01** |  |  | When you work on a computer, it is possible to harm both the computer and yourself. The most common accident that happens when attempting to fix a computer problem is erasing software or data. Knowing and conducting oneself with care can prevent this. |  |
| **SAH-02** | People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally. | Computing devices may be connected to other devices or components to extend their capabilities, such as sensing and sending information. Connections can take many forms, such as physical or wireless. Together, devices and components form a system of interdependent parts that interact for a common purpose. | The interaction between humans and computing devices presents advantages, disadvantages, and unintended consequences. The study of human–computer interaction can improve the design of devices and extend the abilities of humans. | Computing devices are often integrated with other systems, including biological, mechanical, and social systems. These devices can share data with one another. The usability, dependability, security, and accessibility of these devices, and the systems they are integrated with, are important considerations in their design as they evolve. |
| **SAH-03** | A computing system is composed of hardware and software. Hardware consists of physical components, while software provides instructions for the system. These instructions are represented in a form that a computer can understand. | Hardware and software work together as a system to accomplish tasks, such as sending, receiving, processing, and storing units of information as bits. Bits serve as the basic unit of data in computing systems and can represent a variety of information. | Hardware and software determine a computing system’s capability to store and process information. The design or selection of a computing system involves multiple considerations and potential tradeoffs, such as functionality, cost, size, speed, accessibility, and aesthetics. | Levels of interaction exist between the hardware, software, and user of a computing system. The most common levels of software that a user interacts with include system software and applications. |
| **SAH-04** | Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution. | Computing systems share similarities, such as the use of power, data, and memory. Common troubleshooting strategies, such as checking that power is available, checking that physical and wireless connections are working, and clearing out the working memory by restarting programs or devices, are effective for many systems. | Comprehensive troubleshooting requires knowledge of how computing devices and components work and interact. A systematic process will identify the source of a problem, whether within a device or in a larger system of connected devices. | Troubleshooting complex problems involves the use of multiple sources when researching, evaluating, and implementing potential solutions. Troubleshooting also relies on experience, such as when people recognize that a problem is similar to one they have seen before or adapt solutions that have worked in the past. |

| **Skill Identifier** | **Concept** | **Level 1** |
| --- | --- | --- |
| SAH-01 | **Safety** | * Identify ESD (Electrostatic Discharge) precautions and procedures. * Discuss proper disposal of equipment and materials. |
| SAH-02 | **Devices** | * Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use. * Describe how internal and external parts of computing devices function to form a system. * Identify printer technologies, interfaces, and options/upgrades. |
| SAH-03 | **Hardware & Software** | * Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware). * Describe basic hardware and software problems using accurate terminology. * Model how computer hardware and software work together as a system to accomplish tasks * Identify basic concepts and procedures for creating, viewing, and managing disks, directories and files. |
| SAH-04 | **Troubleshooting** | * Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. * Recognize common printer problems and techniques used to resolve them. * Identify techniques of information assurances such as: back-up procedures, data security, storage, offsite and RAID. |

| **Skill Identifier** | **Concept** | **Level 2** |
| --- | --- | --- |
| SAH-01 | **Safety** | * Demonstrate proper tool practices and techniques. * Differentiate between low and high voltage AC and DC electrical systems. * Demonstrate proper care of materials, tools, and equipment. |
| SAH-02 | **Devices** | * Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. |
| SAH-03 | **Hardware & Software** | * Design projects that combine hardware and software components to collect and exchange data. * Describe the major network operating systems and their similarities and differences. |
| SAH-04 | **Troubleshooting** | * Systematically identify and fix problems with computing devices and their components. |

| **Skill Identifier** | **Concept** | **Level 3** |
| --- | --- | --- |
| SAH-01 | **Safety** | * Practice proper electrical safety. |
| SAH-02 | **Devices** | * Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. |
| SAH-03 | **Hardware & Software** | * Compare levels of abstraction and interactions between application software, system software, and hardware layers. |
| SAH-04 | **Troubleshooting** | * Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. |

| **Skill Identifier** | **Concept** | **Level 4** |
| --- | --- | --- |
| SAH-01 | **Safety** |  |
| SAH-02 | **Devices** |  |
| SAH-03 | **Hardware & Software** | * Categorize the roles of operating system software. * Illustrate ways computing systems implement logic, input, and output through hardware components. |
| SAH-04 | **Troubleshooting** |  |

# 

# 

| System Administration (Software) | | | | |
| --- | --- | --- | --- | --- |
| **Skill Identifier** | **Level 1** | **Level 2** | **Level 3** | **Level 4** |
| **SAS-01** |  |  |  |  |
| **SAS-02** |  |  |  |  |
| **SAS-03** |  |  |  |  |
| **SAS-04** |  |  |  |  |

| **Skill Identifier** | **Concept** | **Level 1** |
| --- | --- | --- |
| SAS-01 |  |  |
| SAS-02 |  |  |
| SAS-03 |  |  |
| SAS-04 |  |  |

| **Skill Identifier** | **Concept** | **Level 2** |
| --- | --- | --- |
| SAS-01 |  |  |
| SAS-02 |  |  |
| SAS-03 |  |  |
| SAS-04 |  |  |

| **Skill Identifier** | **Concept** | **Level 3** |
| --- | --- | --- |
| SAS-01 |  |  |
| SAS-02 |  |  |
| SAS-03 |  |  |
| SAS-04 |  |  |

| **Skill Identifier** | **Concept** | **Level 4** |
| --- | --- | --- |
| SAS-01 |  |  |
| SAS-02 |  |  |
| SAS-03 |  |  |
| SAS-04 |  |  |

# 

# 

| Cybersecurity | | | | |
| --- | --- | --- | --- | --- |
| **Skill Identifier** | **Level 1** | **Level 2** | **Level 3** | **Level 4** |
| **CYS-01** |  |  |  |  |
| **CYS-02** |  |  |  |  |
| **CYS-03** |  |  |  |  |
| **CYS-04** |  |  |  |  |

| **Skill Identifier** | **Concept** | **Level 1** |
| --- | --- | --- |
| CYS-01 |  |  |
| CYS-02 |  |  |
| CYS-03 |  |  |
| CYS-04 |  |  |

| **Skill Identifier** | **Concept** | **Level 2** |
| --- | --- | --- |
| CYS-01 |  |  |
| CYS-02 |  |  |
| CYS-03 |  |  |
| CSYS-04 |  |  |

| **Skill Identifier** | **Concept** | **Level 3** |
| --- | --- | --- |
| CYS-01 |  |  |
| CYS-02 |  |  |
| CYS-03 |  |  |
| CYS-04 |  |  |

| **Skill Identifier** | **Concept** | **Level 4** |
| --- | --- | --- |
| CYS-01 |  |  |
| CYS-02 |  |  |
| CYS-03 |  |  |
| CYS-04 |  |  |

# 

# 

| Data Analysis | | | | |
| --- | --- | --- | --- | --- |
| **Skill Identifier** | **Level 1** | **Level 2** | **Level 3** | **Level 4** |
| **DA-01** |  |  |  |  |
| **DA-02** |  |  |  |  |
| **DA-03** |  |  |  |  |
| **DA-04** |  |  |  |  |

| **Skill Identifier** | **Concept** | **Level 1** |
| --- | --- | --- |
| DA-01 |  |  |
| DA-02 |  |  |
| DA-03 |  |  |
| DA-04 |  |  |

| **Skill Identifier** | **Concept** | **Level 2** |
| --- | --- | --- |
| DA-01 |  |  |
| DA-02 |  |  |
| DA-03 |  |  |
| DA-04 |  |  |

| **Skill Identifier** | **Concept** | **Level 3** |
| --- | --- | --- |
| DA-01 |  |  |
| DA-02 |  |  |
| DA-03 |  |  |
| DA-04 |  |  |

| **Skill Identifier** | **Concept** | **Level 4** |
| --- | --- | --- |
| DA-01 |  |  |
| DA-02 |  |  |
| DA-03 |  |  |
| DA-04 |  |  |

# 

# 

# Example Classes

| Physical Programming Physical Computing with Arduino allows students to refresh their knowledge of basic programming concepts (control structures, variables, functions, etc.) in order to control a physical device. Students will perform basic physical tasks using LEDs, motors, and sensors to see how computer programming gives physical devices the ability to interact with their environment.  Quarter Long Elective || Semester Long | | |
| --- | --- | --- |
| **Competency** | **\*End of class Assessment** | **Skill Identifiers being Tested** |
| Algorithms and Programming | Level 3 | AP-01  AP-02  AP-03  AP-04  AP-05 |
| Computing Systems | Level 2-3 | CS-01  CS-02  CS-03  CS-04 |
| Career Ready | \*\*Level 2 | CR-01  CR-02  \*\* CR-04 |

\*Where students should test out Level wise for the competencies

\*\* I would say higher if you have a portfolio project or project based approach

| Cybersecurity Cybersecurity affects every individual, organization, and nation. This course focuses on the evolving and pervasive technological environment with an emphasis on securing personal, organizational, and national information. Students will be introduced to the principles of cybersecurity, explore emerging technologies, examine threats and protective measures, and investigate the diverse high-Skill Identifier, high-wage, and high-demand career opportunities in the field of cybersecurity.  Quarter Long Elective | | |
| --- | --- | --- |
| **Competency** | **End of class Assessment** | **Skill Identifiers being Tested** |
| Algorithms and Programming | Level 2 | AP-01  AP-03  AP-04 |
| Networks and the Internet | Level 3 | NI-01  NI-02  NI-03 |
| Computing Systems | Level 3 | CS-01  CS-02  CS-03  CS-04 |
| Career Ready | Level 2 | CR-02  CR-03 |

| AP Computer Science A AP Computer Science A introduces students to computer science through programming. Fundamental topics in this course include the design of solutions to problems, the use of data structures to organize large sets of data, the development and implementation of algorithms to process data and discover new information, the analysis of potential solutions, and the ethical and social implications of computing systems. The course emphasizes object-oriented programming and design using the Java programming language.  Semester Long || Year Long | | |
| --- | --- | --- |
| **Competency** | **End of class Assessment** | **Skill Identifiers being Tested** |
| Algorithms and Programming | Level 3-4 | AP-01  AP-02  AP-03  AP-04  AP-05 |
| Career Ready | Level 3 | CR-01  CR-02  CR-03  CR-04 |