CS.1 Development

I can innovate through a user-focused iterative design process, integrating implementation/feedback cycles. I value experimentation and risk-taking, fostering a creative environment for continuous improvement.

Criteria	Proficient	Example
CS.1.1 **Define Program Function**	- Accurately defines the purpose and functionality of a program.	Clearly articulating that a payroll program calculates employee salaries based on hours worked and hourly rates, providing a concise overview of its primary function.
CS.1.2 **Program Design**	- Creates a well-structured program design for a given task.	Creating a structured program design for a simple task, such as designing a program to calculate and display the average of a list of numbers.
CS.1.3 **Identifying and Correcting Errors**	- Recognizes and corrects common errors in a given program.	Recognizing and fixing syntax errors in a program, such as resolving issues with missing semicolons or incorrect variable names.
CS.1.4 **Working in Teams**	- Effectively collaborates and communicates within a team.	Collaborating effectively within a programming team, communicating ideas, and contributing to the development of a shared project, showcasing teamwork skills.
CS.1.5 **Development Cycle**	- Understands and applies the basic concepts of the development cycle.	Understanding and applying the development cycle by following a systematic approach, including planning, coding, testing, and debugging, in the creation of a small program.
CS.1.6 **Documentation**	- Produces clear and organized documentation for a program.	Creating clear and organized documentation for a program, including comments within the code and an external readme file explaining program functionality and usage.

CS.1.7 **User Testing**	- Conducts basic user testing to assess program functionality.	Conducting basic user testing by having individuals use the program and providing feedback, allowing for identification of usability issues and areas for improvement.
Last edited:	12/13/2023	

CS.2 Programming

I comprehend the significance of variables, data abstraction, expressions, strings, conditionals, iteration, procedures, and libraries in programming. Integrating random values and objects fosters creative and effective problem-solving for robust software solutions.

Criteria	Proficient	Example
CS.2.1 **Variables and Assignments**	- Successfully uses variables and assignments in programming.	Declaring and assigning values to variables, such as initializing a variable counter to zero and updating it within a loop.
CS.2.2 **Data Abstraction**	- Applies data abstraction techniques to simplify program design.	Using a function to abstract details of data storage, simplifying the main program logic by encapsulating complex data operations.
CS.2.3 **Mathematical Expressions**	- Constructs and evaluates mathematical expressions accurately.	Creating and evaluating mathematical expressions, like calculating the area of a rectangle using the formula area = length * width.
CS.2.4 **Strings**	- Manipulates strings effectively within a programming context.	Manipulating strings by concatenating them, extracting substrings, or converting case, demonstrating proficiency in string operations.
CS.2.5 **Boolean Expressions**	- Constructs and evaluates boolean expressions correctly.	Constructing and evaluating boolean expressions, such as using logical operators (AND, OR) to make decisions in a program.
CS.2.6 **Conditionals**	- Implements conditional statements to control program flow.	Implementing conditional statements, like an if-else statement to check if a variable is greater than a threshold and execute corresponding code.
CS.2.7 **Iteration**	- Utilizes iteration constructs to repeat code execution.	Utilizing a for or while loop to iterate over a list, demonstrating the ability to repeat a block of code multiple times.

CS.2.8 **Procedures**	- Defines and calls procedures to modularize program structure.	Defining a procedure to calculate the average of a list of numbers, then calling it within the main program to modularize the code.
CS.2.9 **Libraries**	- Incorporates external libraries effectively in programming.	Incorporating an external library like numpy in Python to efficiently perform array operations, showcasing effective library usage.
CS.2.10 **Random Values**	- Generates and uses random values appropriately in code.	Generating a random number within a specified range and using it in a program, such as simulating a dice roll in a game.
CS.2.11 **Objects**	- Demonstrates an understanding of object-oriented programming concepts.	Demonstrating understanding of object-oriented programming by creating and using objects, like defining a Car object with attributes and methods.
Last edited:	12/13/2023	

CS.3 Algorithms

I grasp essential computer science concepts like algorithm development, lists, searches, sorts, and algorithmic efficiency. I also understand undecidable problems and the impact of parallel and distributed computing in modern environments.

Criteria	Proficient	Example
CS.3.1 **Developing Algorithms**	- Designs algorithms to solve specific problems with attention to efficiency.	Designing an algorithm to sort a collection of data efficiently, considering time and space complexity constraints.
CS.3.2 **Lists / arrays**	- Implements algorithms for basic list/array manipulation and retrieval.	Implementing an algorithm to search and retrieve specific elements from a list or array based on predefined criteria.
CS.3.3 **Searches**	- Creates fundamental search algorithms for efficient data retrieval.	Creating a binary search algorithm for a sorted list, showcasing proficiency in fundamental search strategies.
CS.3.4 **Sorts**	- Implements simple sorting algorithms and compares their performance.	Implementing a bubble sort algorithm and comparing its performance to other simple sorting algorithms.
CS.3.5 **Simulations**	- Develops basic simulations to model and understand simple systems.	Developing a basic simulation to model the behavior of a population over time, demonstrating an understanding of simulation concepts.
CS.3.6 **Algorithmic Efficiency**	- Analyzes and proposes simple optimizations for algorithms.	Analyzing the time complexity of a given algorithm and proposing simple optimizations to enhance its efficiency.

CS.3.7 **Undecidable Problems**	- Explains undecidable problems and foundational theoretical concepts.	Explaining the concept of the Halting Problem and its undecidability, demonstrating a foundational understanding of theoretical concepts in computer science.
CS.3.8 **Parallel and Distributed Computing**	- Implements basic parallelized algorithms on multi-core processors.	Implementing a basic parallelized algorithm for a straightforward computation on a multi-core processor, showcasing an understanding of basic parallel computing concepts.
Last edited:	12/13/2023	

CS.4 Data

I proficiently apply notation systems, data compression, data extraction, program data usage, and database fundamentals. This includes representing data, efficient compression, information extraction, program-based data manipulation, and understanding database fundamentals.

Criteria	Proficient	Example
CS.4.1 **Notation Systems**	- Successfully interprets and uses notation systems in computer science.	Successfully interpreting and applying binary notation in programming, explaining the representation of numbers, and creating a binary-to-decimal conversion tool.
CS.4.2 **Data Compression**	- Understands and applies basic data compression techniques effectively.	Implementing a basic data compression algorithm, analyzing its effectiveness, and presenting findings.
CS.4.3 **Extracting Information from Data**	- Demonstrates the ability to extract relevant information from structured and unstructured data.	Analyzing a dataset, extracting relevant information, and presenting insights.
CS.4.4 **Using Programs with Data**	- Effectively integrates data into programs, demonstrating proficiency in data utilization.	Integrating data into a program, manipulating and analyzing it to produce meaningful outputs, and presenting the results.
CS.4.5 **Using Programs with Data Visualization**	- Utilizes programs to create effective data visualizations for improved understanding.	Creating data visualizations using programming tools, explaining design choices, and presenting insights gained through visualizations.

CS.4.6 **Database Fundamentals**	- Grasps fundamental concepts of databases, including design and query execution.	Designing and implementing a simple relational database, creating SQL queries, and presenting findings.
Last edited:	12/13/2023	

CS.5 Systems

I adeptly analyze data to comprehend my surroundings and recognize the impact of my actions, contributing to a broader understanding of data's significance in various contexts.

Criteria	Proficient	Example
CS.5.1 Computer Hardware	- Demonstrates a solid understanding of computer hardware components and their functions.	Successfully assembling and upgrading computer components, explaining their functions and interactions, and troubleshooting hardware issues.
CS.5.2 Networking Concepts	- Grasps fundamental networking concepts, including protocols, addressing, and communication.	Designing and implementing a small network, configuring devices, explaining the purpose of networking protocols, and troubleshooting connectivity issues.
CS.5.3 Software	- Understands the basics of software, including types, functions, and the software development process.	Developing a small software application, explaining the software development life cycle, and troubleshooting software bugs.
CS.5.4 Computer Interfacing	- Proficiently interfaces with computer systems, understanding input/output devices and user interaction.	Successfully connecting and configuring diverse peripherals, explaining input/output devices, and troubleshooting interface issues.
CS.5.5 Internet of Things	- Understands the concept of the Internet of Things (IoT) and its applications in connecting devices.	Implementing a small IoT project, connecting devices, explaining communication protocols, and troubleshooting connectivity in an IoT system.
CS.5.6 Troubleshooting	- Applies effective troubleshooting techniques to identify and resolve common computer and network issues.	Effectively troubleshooting a network issue, identifying and resolving the root cause, documenting the process, and explaining the solution.
CS.5.7 Cybersecurity	- Grasps fundamental cybersecurity concepts, including basic principles of protecting systems and data.	Developing and implementing a basic cybersecurity plan, explaining principles of data protection, and participating in a simulated cyberattack response.
Last edited:	12/13/2023	

CS.6 Impacts

I grasp the dual impact of widespread computer use on the environment, society, and individuals. Identifying harm, supporting mitigation, and addressing societal changes, misinformation, mental health, ethics, and security are crucial.

Criteria	Proficient	Example
CS.6.1 **Technology and the Environment**	- Demonstrates an understanding of the impact of technology on the environment, including sustainability considerations.	Investigating and presenting the environmental impact of a specific technology, considering sustainability factors, and proposing actionable steps for responsible usage and disposal.
CS.6.2 **Technology and Society**	- Understands the relationship between technology and society, recognizing the influence of technology on social structures and behaviors.	Analyzing the societal impact of a technological advancement, discussing its implications on social structures, and presenting findings to peers, showcasing a solid understanding.
CS.6.3 **Ethics and Security**	- Grasps fundamental ethical principles related to technology use and understands the importance of cybersecurity.	Actively participating in a cybersecurity workshop, discussing ethical considerations, and demonstrating the application of secure practices in protecting digital information.
CS.6.4 **Artificial Intelligence**	- Recognizes the basics of artificial intelligence and its applications, including ethical considerations in AI development and use.	Developing a project that involves applying ethical principles in AI design, considering issues like bias and transparency, and presenting the project to peers.
Last edited:	12/13/2023	