| Self-Directed Learner |  | |  |  | |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Level of Mastery** | **Doesn’t Understand Yet** | | **Uncertain** | **Gets It** | | **State & National Standards** |
| **Competency** | 0 – 1: No Evidence | 2: Not Yet | 3 – 4: Developing | 5 – 6: Proficient | 7 – 8: Exemplary |
| Identifies new information needed to complete a task. |  |  | Identifies, possibly with support, knowledge and skill needed to accomplish a goal. | Independently identifies necessary knowledge and skill. |  |  |
| Selecting and reading complex informational text to accomplish a goal. | Uses supportive text with assistance. | Reads assigned supportive text and demonstrates most of the skills. | Reads teacher selected authentic texts and successfully demonstrates the skills and/or knowledge contained. [material is provided] | Identifies desired information and obtains it by searching for and finding an appropriate source which is read and applied to the problem. [identifies target and finds and uses a single source] | Identifies desired information and obtains it by searching for and finding a variety of materials, reading them, and synthesizing the information into a useable body of knowledge. [combines from multiple sources] |  |
| Takes notes | Sporadic note taking. Little organization and notes are often incomplete. | Takes notes when directed to.  Content largely reflects content of presentation. | Independent note taking is beginning to emerge – basic structures of organization are generally present. | Self-motivated to take notes.  Effort is put into organizing and structuring notes.  Evidence is present that notes are being used as a strategy for problem solving. | A variety of note taking purposes and strategies are applied.  Notes are reviewed and revised and/or extended. |  |
| Effectively seeks help with clarifying understanding |  |  | Asks questions when needed, with help can identify a specific problem area or question. | Asks clear and specific questions. | Seeks and uses outside sources to resolve questions.  Asks clear and well received questions.  Maintains relationships with respondents. |  |
| Tests knowledge |  |  |  |  | Critically and accurately evaluates understanding and uses the result as feedback into the process. |  |

| Professionalism |  | |  |  | |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Level of Mastery** | **Doesn’t Understand Yet** | | **Uncertain** | **Gets It** | | **State & National Standards** |
| **Competency** | 0 – 1: No Evidence | 2: Not Yet | 3 – 4: Developing | 5 – 6: Proficient | 7 – 8: Exemplary |
| Ethics & responsibilities |  |  |  |  |  |  |
| Working with diverse teams |  |  |  |  |  |  |
| Leadership |  |  |  |  |  |  |
| Developing and maintaining a professional online presence |  |  |  |  |  |  |
| Understanding of the career pathway |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Using Digital Tools |  | |  |  | |  |
| Design Process |  | |  |  | |  |
| **Level of Mastery** | **Doesn’t Understand Yet** | | **Uncertain** | **Gets It** | | **State & National Standards** |
| **Competency** | 0 – 1: No Evidence | 2: Not Yet | 3 – 4: Developing | 5 – 6: Proficient | 7 – 8: Exemplary |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

| Computer Science |  | |  |  | |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Level of Mastery** | **Doesn’t Understand Yet** | | **Uncertain** | **Gets It** | | **State & National Standards** |
| **Competency** | 0 – 1: No Evidence | 2: Not Yet | 3 – 4: Developing | 5 – 6: Proficient | 7 – 8: Exemplary |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

| Project Management |  | |  |  | |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Level of Mastery** | **Doesn’t Understand Yet** | | **Uncertain** | **Gets It** | | **State & National Standards** |
| **Competency** | 0 – 1: No Evidence | 2: Not Yet | 3 – 4: Developing | 5 – 6: Proficient | 7 – 8: Exemplary |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

| **Level of Mastery** | **Doesn’t Understand Yet** | | **Uncertain** | **Gets It** | | **State & National Standards** |
| --- | --- | --- | --- | --- | --- | --- |
| **Competency** | 0 – 1: No Evidence | 2: Not Yet | 3 – 4: Developing | 5 – 6: Proficient | 7 – 8: Exemplary |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

ECS Computational Practices:

* Analyze the effects of developments in computing (impact/connections)
* Design and implement creative solutions and artifacts
* Apply abstractions and models
* Analyze their computational work and the work of others
* Communicate computational thought processes, procedures, and results to others
* Collaborate with peers on computing activities

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | CCSS.ELA-Literacy.CCRA.L.6 | Acquire and use accurately a range of general academic and domain-specific words and phrases sufficient for reading, writing, speaking, and listening at the college and career readiness level; demonstrate independence in gathering vocabulary knowledge when encountering an unknown term important to comprehension or expression. |  |
|  | CCSS.ELA-Literacy.CCRA.R.2 | Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas. |  |
|  | CCSS.ELA-Literacy.CCRA.SL.2 | Integrate and evaluate information presented in diverse media and formats, including visually, quantitatively, and orally. |  |
|  | CCSS.ELA-Literacy.CCRA.W.10 | Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of tasks, purposes, and audiences. |  |
|  | CCSS.ELA-Literacy.CCRA.W.6 | Use technology, including the Internet, to produce and publish writing and to interact and collaborate with others. |  |
|  | CCSS.ELA-Literacy.CCRA.W.7 | Conduct short as well as more sustained research projects based on focused questions, demonstrating understanding of the subject under investigation. |  |
|  | CCSS.ELA-Literacy.CCRA.W.8 | Gather relevant information from multiple print and digital sources, assess the credibility and accuracy of each source, and integrate the information while avoiding plagiarism. |  |
|  | CCSS.Math.Practice.MP1 | Make sense of problems and persevere in solving them. |  |
|  | CCSS.Math.Practice.MP2 | Reason abstractly and quantitatively. |  |
|  | CCSS.Math.Practice.MP3 | Construct viable arguments and critique the reasoning of others. |  |
|  | CCSS.Math.Practice.MP4 | Model with Mathematics. |  |
|  | CCSS.Math.Content.HSF-BF.A.1a | Building Functions - Write a function that describes a relationship between two quantities: Determine an explicit expression, a recursive process, or steps for calculation from a context. |  |
|  | CCSS.Math.Content.HSS-CP.A.1 | Conditional Probability and the Rules of Probability - Understand independence and conditional probability and use them to interpret data: Describe events as subsets of a sample space (the set of outcomes) using characteristics (or  categories) of the outcomes, or as unions, intersections, or complements of other events ("or" and "not"). |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Technology Literacy** is the ability to responsibly, creatively and effectively use appropriate technology to:

\* Communicate and collaborate.

\* Access, collect, manage, integrate and evaluate information.

\* Solve problems and create solutions

\* Build and share knowledge.

\* Improve and enhance learning in all subject areas and experiences.

**Technology Fluency** is demonstrated when students:

\* Apply technology to real-world experiences.

\* Adapt to changing technologies.

\* Modify current and create new technologies.

\* Personalize technology to meet personal needs, interests and learning style

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| EL | **1. Empowered Learner** - Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences. |  |  |
| DC | **2. Digital Citizen** - Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical. |  |  |
| KC | **3. Knowledge Constructor** - Students critically curate a variety of resources using digital tools to construct knowledge, produce creative artifacts and make meaningful learning experiences for themselves and others. |  |  |
| ID | **4. Innovative Designer** - Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions. |  |  |
| CT | **5. Computational Thinker** - Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions. |  |  |
| CC | **6. Creative Communicator** - Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals. |  |  |
| GC | **7. Global Collaborator** - Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally. |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Practices / Concepts | 1. Computing Systems | 2. Networks and the Internet | 3. Data and Analysis | 4. Algorithms and Programming | 5. Impacts of Computing |
| 1. Fostering an Inclusive Computing Culture |  |  |  |  |  |
| 2. Collaborating Around Computing |  |  |  |  |  |
| 3. Recognizing and Defining Computational Problems |  |  |  |  |  |
| 4. Developing and Using Abstractions |  |  |  |  |  |
| 5. Creating Computational Artifacts |  |  |  |  |  |
| 6. Testing and Refining Computational Artifacts |  |  |  |  |  |
| 7. Communicating About Computing |  |  |  |  |  |

| Identifier Code | C | P | 9–10: Level 3A |
| --- | --- | --- | --- |
| 3A-I-1-26 | I | 1 | Compare and debate the positive and negative impacts of computing on behavior and culture (e.g., evolution from hitchhiking to ridesharing apps, online accommodation rental services). |
| 3A-I-1-27 | I | 1 | Demonstrate how computing enables new forms of experience, expression, communication, and collaborating. |
| 3A-I-1-28 | I | 1 | Explain the impact of the digital divide (i.e., uneven access to computing, computing education, and interfaces) on access to critical information. |
| 3A-N-1-32 | N | 1 | Compare and contrast multiple viewpoints on cybersecurity (e.g., from the perspective of security experts, privacy advocates, the government). |
| 3A-A-2-1 | A | 2 | Design and develop a software artifact working in a team. |
| 3A-A-2-2 | A | 2 | Demonstrate how diverse collaborating impacts the design and development of software products (e.g., discussing real-world examples of products that have been improved through having a diverse design team or reflecting on their own team's development experience). |
| 3A-I-2-22 | I | 2 | Debate the social and economic implications associated with ethical and unethical computing practices (e.g., intellectual property rights, hacktivism, software piracy, diesel emissions testing scandal, new computers shipped with malware). |
| 3A-A-3-10 | A | 3 | Design algorithms using sequence, selection, and iteration. |
| 3A-A-3-11 | A | 3 | Explain and demonstrate how modeling and simulation can be used to explore natural phenomena (e.g., flocking behaviors, queueing, life cycles). |
| 3A-D-3-20 | D | 3 | Discuss techniques used to store, process, and retrieve different amounts of information (e.g., files, databases, data warehouses). |
| 3A-D-3-21 | D | 3 | Apply basic techniques for locating and collecting small- and large-scale data sets (e.g., creating and distributing user surveys, accessing real-world data sets). |
| 3A-N-3-33 | N | 3 | Explain the principles of information security (confidentiality, integrity, availability) and authentication techniques. |
| 3A-N-3-34 | N | 3 | Use simple encryption and decryption algorithms to transmit/receive an encrypted message. |
| 3A-A-4-7 | A | 4 | Understand the notion of hierarchy and abstraction in high-level languages, translation, instruction sets, and logic circuits. |
| 3A-A-4-8 | A | 4 | Deconstruct a complex problem into simpler parts using predefined constructs (e.g., functions and parameters and/or classes). |
| 3A-A-4-9 | A | 4 | Demonstrate the value of abstraction for managing problem complexity (e.g., using a list instead of discrete variables). |
| 3A-C-4-15 | C | 4 | Demonstrate the role and interaction of a computer embedded within a physical system, such as a consumer electronic, biological system, or vehicle, by creating a diagram, model, simulation, or prototype. |
| 3A-C-4-16 | C | 4 | Describe the steps necessary for a computer to execute high compilation to machine language, interpretation, fetch-decode-execute <https://www.cise.ufl.edu/~mssz/CompOrg/CDAintro.html>. |
| 3A-D-4-18 | D | 4 | Convert between binary, decimal, and hexadecimal representations of data (e.g., convert hexadecimal color codes to decimal percentages, ASCII/Unicode representation). |
| 3A-D-4-19 | D | 4 | Analyze the representation tradeoffs among various forms of digital information (e.g., lossy versus lossless compression, encrypted vs. unencrypted, various image representations). |
| 3A-N-4-31 | N | 4 | Illustrate the basic components of computer networks (e.g., draw logical and topological diagrams of networks including routers, switches, servers, and end user devices; create model with string and paper). |
| 3A-A-5-4 | A | 5 | Design, develop, and implement a computing artifact that responds to an event (e.g., robot that responds to a sensor, mobile app that responds to a text message, sprite that responds to a broadcast). |
| 3A-A-5-5 | A | 5 | Use user-centered research and design techniques (e.g., surveys, interviews) to create software solutions |
| 3A-A-5-6 | A | 5 | Integrate grade-level appropriate mathematical techniques, concepts, and processes in the creation of computing artifacts. |
| 3A-C-5-14 | C | 5 | Create, extend, or modify existing programs to add new features and behaviors using different forms of inputs and outputs (e.g., inputs such as sensors, mouse clicks, data sets; outputs such as text, graphics, sounds). |
| 3A-D-5-17 | D | 5 | Create computational models that simulate real-world systems (e.g., ecosystems, epidemics, spread of ideas). |
| 3A-A-6-12 | A | 6 | Use a systematic approach and debugging tools to independently debug a program (e.g., setting breakpoints, inspecting variables with a debugger). |
| 3A-I-6-29 | I | 6 | Redesign user interfaces (e.g., webpages, mobile applications, animations) to be more inclusive, accessible, and minimizing the impact of the designer's inherent bias. |
| 3A-N-6-35 | N | 6 | Identify digital and physical strategies to secure networks and discuss the tradeoffs between ease of access and need for security. |
| 3A-A-7-3 | A | 7 | Compare and contrast various software licensing schemes (e.g., open source, freeware, commercial). |
| 3A-C-7-13 | C | 7 | Develop and apply criteria (e.g., power consumption, processing speed, storage space, battery life, cost, operating system) for evaluating a computer system for a given purpose (e.g., system specification needed to run a game, web browsing, graphic design or video editing). |
| 3A-I-7-23 | I | 7 | Compare and contrast information access and distribution rights. |
| 3A-I-7-24 | I | 7 | Discuss implications of the collection and large-scale analysis of information about individuals (e.g., how businesses, social media, and government collect and use personal data). |
| 3A-I-7-25 | I | 7 | Describe how computation shares features with art and music by translating human intention into an artifact. |
| 3A-N-7-30 | N | 7 | Describe key protocols and underlying processes of Internet-based services (e.g., http/https and SMTP/IMAP, routing protocols). |

| 11–12 | **Level 3B** |
| --- | --- |
| 3B-A-2-1 | Use version control systems, integrated development environments (IDEs), and collaborating tools and practices (code documentation) in a group software project. |
| 3B-A-2-2 | Demonstrate software life cycle processes (e.g., spiral, waterfall) by participating on software project teams (e.g., community service project with real-world clients). |
| 3B-A-7-3 | Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality). |
| 3B-A-7-4 | Explain security issues that might lead to compromised computer programs (e.g., circular references, ambiguous program calls, lack of error checking and field size checking). |
| 3B-A-7-5 | Compare a variety of programming languages and identify features that make them useful for solving different types of problems and developing different kinds of systems (e.g., declarative, logic, parallel, functional, compiled, interpreted, real-time). |
| 3B-A-7-6 | Describe how artificial intelligence drives many software and physical systems (e.g., autonomous robots, computer vision, pattern recognition, text analysis). |
| 3B-A-5-7 | Decompose a problem by creating new data types, functions, or classes. |
| 3B-A-5-8 | Demonstrate code reuse by creating programming solutions using libraries and APIs (e.g., graphics libraries, maps API). |
| 3B-A-5-9 | Implement an AI algorithm to play a game against a human opponent or solve a problem. |
| 3B-A-5-10 | Develop programs for multiple computing platforms (e.g., computer desktop, web, mobile). |
| 3B-A-4-11 | Critically analyze classic algorithms (e.g., sorting, searching) and use in different contexts, adapting as appropriate. |
| 3B-A-4-12 | Evaluate algorithms (e.g., sorting, searching) in terms of their efficiency, correctness, and clarity. |
| 3B-A-4-13 | Compare and contrast fundamental data structures and their uses (e.g., lists, maps, arrays, stacks, queues, trees, graphs). |
| 3B-A-4-14 | Discuss issues that arise when breaking large-scale problems down into parts that must be processed simultaneously on separate systems (e.g., cloud computing, parallelization, concurrency). |
| 3B-A-3-15 | Provide examples of computationally solvable problems and difficult-to-solve problems. |
| 3B-A-3-16 | Explain the value of heuristic algorithms (discovery methods) to approximating solutions for difficult-to-solve computational problems. |
| 3B-A-3-17 | Decompose a large-scale computational problem by identifying generalizable patterns and applying them in a solution. |
| 3B-A-3-18 | Illustrate the flow of execution of a recursive algorithm. |
| 3B-A-3-19 | Describe how parallel processing can be used to solve large problems (e.g., SETI at Home, FoldIt). |
| 3B-A-3-20 | Develop and use a series of test cases to verify that a program performs according to its design specifications. |
| 3B-A-6-21 | Evaluate key qualities of a program (e.g., correctness, usability, readability, efficiency, portability, scalability) through a process such as a code review. |
| 3B-C-7-22 | Explain the role of operating systems (e.g., how programs are stored in memory, how data is organized/retrieved, how processes are managed and multi-tasked). |
| 3B-C-7-23 | Identify the functionality of various categories of hardware components and communication between them (e.g., physical layers, logic gates, chips, input and output devices). |
| 3B-D-4-24 | Use data analysis to identify significant patterns in complex systems (e.g., take existing data sets and make sense of them). |
| 3B-D-4-25 | Discuss how data sequences (e.g., binary, hexadecimal, octal) can be interpreted in a variety of forms (e.g., instructions, numbers, text, sound, image). |
| 3B-D-4-26 | Evaluate the ability of models and simulations to formulate, refine, and test hypotheses. |
| 3B-D-4-27 | Identify mathematical and computational patterns through modeling and simulation (e.g., regression, Runge-Kutta, queueing theory, discrete event simulation). |
| 3B-D-1-28 | Use various data collection techniques for different types of problems (e.g., mobile device, GPS, user surveys, embedded system sensors, open data sets, social media data sets). |
| 3B-D-3-29 | Explore security policies by implementing and comparing encryption and authentication strategies (e.g., secure coding, safeguarding keys). |
| 3B-I-7-30 | Develop criteria to evaluate the beneficial and harmful effects of computing innovations on people and society. |
| 3B-I-5-31 | Select, observe, and contribute to global Collaborating in the development of a computational artifact (e.g., contribute the resolution of a bug in an open-source project hosted on GitHub). |
| 3B-I-1-32 | Design and implement a study that evaluates or predicts how computation has revolutionized an aspect of our culture and how it might evolve (e.g., education, healthcare, art/entertainment, energy). |
| 3B-I-1-33 | Debate laws and regulations that impact the development and use of software. |
| 3B-I-1-34 | Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society. |
| 3B-N-4-35 | Simulate and discuss the issues (e.g., bandwidth, load, delay, topology) that impact network functionality (e.g., use free network simulators). |