| **Concepts** | **Practices** | | | | | | | **Level** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1: Fostering an Inclusive Computing Culture** | **2: Collaborating** | **3: Recognizing & Defining Computational Problems** | **4: Developing & Using Abstractions** | **5: Creating Computational Artifacts** | **6: Testing & Refining** | **7: Communicating about Computing** |
| **A: Algorithms & Programming** |  | **1:** Solicit and integrate peer feedback as appropriate to develop or refine a program. | **9:** Decompose a problem into parts & create solutions for each part. | **8:** Define & use procedures that hide the complexity of a task & can be reused to solve similar tasks. | **5:** Design, develop, & presentcomputational artifacts both independently & collaboratively.  **6:** Develop programs, both independently & collaboratively, that include sequences with nested loops & multiple branches.  **7:** Create variables thot represent different data types & manipulate them. | **10:** Use an iterative design process to solve problems, both independently & collaboratively. | **2:** Compare different algorithms that may be used to solve the same problem.  **3:** Provide proper attribution when code is borrowed or built upon.  **4:** Interpret the flow of execution of algorithms and predict their outcomes. | **2** |
|  | **1:** Design & develop a software artifact working in a team.  **2:** Demonstrate how diverse collaborating impacts the design & development of software products. | **10:** Design algorithms using sequence, selection, and iteration.  **11:** Explain & demonstrate how modeling & simulation can be used to explore natural phenomena. | **7:** Understand the notion of hierarchy & abstraction in high-level languages, translation, instruction sets, & logic circuits.  **8:** Deconstruct a complex problem into simpler parts using predefined constructs.  **9:** Demonstrate the value of abstraction for managing problem complexity. | **4:** Design, develop, & implement a computing artifact that responds to an event.  **5:** Use user-centered research & design techniques to create software solutions.  **6:** Integrate grade-level appropriate mathematical techniques, concepts, & processes into the creation of computing artifacts. | **12:** Use a systematic approach to independently debug a program. | **3:** Compare & contrast various software license schemes. | **3A** |
|  | **1:** Use version control systems, integrated development environments (IDEs), and collaboration tools and practices in a group software project.  **2:** Demonstrate software life cycle processes by participating on software project teams. | **15:** Provide examples of computationally solvable problems & of difficult to solve problems.  **16:** Explain the value of heuristic algorithms in approximating solutions for difficult-to-solve computational problems.  **17:** Decompose a large-scale computational problem by identifying generalizable patterns & using them in a solution.  **18:** Illustrate the flow of execution in a recursive algorithm.  **19:** Describe how parallel processing can be used to solve large problems.  **20:** Develop & use a series of test cases to verify that a program performs according to its design specifications. | **11:** Critically analyze classic algorithms and use them in different contexts, adapting as appropriate.  **12:** Evaluate algorithms in terms of their efficiency, correctness, & clarity.  **13:** Compare & contrast fundamental data structures & their uses.  **14:** Discuss issues that arise when breaking large-scale problems into parts that must be processed simultaneously on separate systems. | **7:** Decompose a problem by creating new data types, functions, or classes.  **8:** Demonstrate code reuse by creating programming solutions using libraries and APIs.  **9:** Implement an AI algorithm to play a game against a human opponent or solve a problem.  **10:** Develop programs for multiple computing platforms. | **21:** Evaluate key qualities of a program through a process such as code review. | **3:** Modify an existing program to add additional functionality & discuss intended & unintended implications.  **4:** Explain security issues that might lead to compromised computer programs.  **5:** Compare a variety of programming languages & identify features that make them useful for solving different types of problems & developing different kinds of systems.  **6:** Describe how AI drives many software & physical systems. | **3B** |
| **C: Computing Systems** |  |  |  | **12:** Analyze the relationship between a device’s computational components & its capabilities. |  | **13:** Use a systematic process to identify the source of problems within individual & interconnected devices. | **11:** Justify the hardware & software chosen to accomplish a task. | **2** |
|  |  |  | **15:** Demonstrate the role & interaction of a computer embedded within a physical system.  **16:** Describe the steps necessary for a computer to execute code written in a high-level language. | **14:** Create, extend, or modify existing programs to add new features & behaviors using different forms of inputs & outputs. |  | **13:** Develop & apply criteria for evaluating a computer system for a given purpose. | **3A** |
|  |  |  |  |  |  |  | **3B** |
| **D: Data & Analysis** |  |  |  |  |  |  |  | **2** |
|  |  | **20:** Discuss techniques used to store, process, & retrieve different amounts of information.  **21:** Apply basic techniques for locating & collecting small- & large-scale data sets. | **18:** Convert between binary, decimal, & hexadecimal representations of data.  **19:** Analyze the representation tradeoffs between various forms of digital information. | **17:** Create computational models that simulate real-world systems. |  |  | **3A** |
|  |  |  |  |  |  |  | **3B** |
| **I: Impacts of Computing** | **20:** Provide examples of how computational artifacts impact health & wellbeing, both positively & negatively.  **21:** Describe ways in which the Internet impacts global communication and collaboration.  **22:** Describe ethical issues that relate to computing devices & networks. |  |  |  |  | **23:** Redesign a computational artifact to remove barriers to universal access. | **18:** Summarize negative & positive impacts of using data & information to categorize people, predict behavior, & make recommendations based on those predictions.  **19:** Explain how computer science fosters innovation & enhances nearly all careers & disciplines. | **2** |
| **26:** Demonstrate how computing enables new forms of experience, expression, communication, & collaboration.  **27:** Explain the impact of the digital divide on access to critical information. | **22:** Debate the social & economic implications associated with ethical & unethical computing practices. |  |  |  | **29:** Redesign user interfaces to be more inclusive, accessible, & to minimize the impact of the designer’s inherent bias. | **23:** Compare & contrast information access & distribution rights.  **24:** Discuss implications of the collection & large-scale analysis of information about individuals.  **25:** Describe how computing shares features with art & music by translating human intention into an artifact. | **3A** |
|  |  |  |  |  |  |  | **3B** |
| **N: Networks & the Internet** |  |  |  |  |  |  |  | **2** |
|  |  |  |  |  |  |  | **3A** |
|  |  |  |  |  |  |  | **3B** |