**AP Computer Science Principles: Des Moines Public Schools**

2017-18 CURRICULUM GUIDE TEC551/552

|  |
| --- |
| **AP Computer Science Principles** |
| The AP Computer Science Principles course is designed to be equivalent to a first-semester introductory college computing course. In this course, students will develop computational thinking vital for success across all disciplines, such as using computational tools to analyze and study data and working with large data sets to analyze, visualize, and draw conclusions from trends. The course is unique in its focus on fostering student creativity. Students are encouraged to apply creative processes when developing computational artifacts and to think creatively while using computer software and other technology to explore questions that interest them. They will also develop effective communication and collaboration skills, working individually and collaboratively to solve problems, and discussing and writing about the importance of these problems and the impacts to their community, society, and the world.  **AP Computer Science Principles**  **– Course Content:**  The following are the major areas of study, or big ideas which are foundational to studying computer science:  • **Creativity**: Computing is a creative activity. Creativity and computing are prominent forces in innovation; the innovations enabled by computing have had and will continue to have far-reaching impact.  • **Abstraction**: Abstraction reduces information and detail to facilitate focus on relevant concepts. It is a process, a strategy, and the result of reducing detail to focus on concepts relevant to understanding and solving problems.  • **Data and Information**: Data and information facilitate the creation of knowledge. Computing enables and empowers new methods of information processing, driving monumental change across many disciplines — from art to business to science.  • **Algorithms**: Algorithms are used to develop and express solutions to computational problems. Algorithms realized in software have affected the world in profound and lasting ways.  • **Programming**: Programming enables problem solving, human expression, and creation of knowledge. Programming and the creation of software has changed our lives. Programming results in the creation of software, and it facilitates the creation of computational artifacts, including music, images, and visualizations.  • **The Internet**: The Internet pervades modern computing. The Internet and the systems built on it have had a profound impact on society. Computer networks support communication and collaboration.  • **Global Impact**: Computing has global impact. Our methods for communicating, collaborating, problem solving, and doing business have changed and are changing due to innovations enabled by computing.  **AP Computer Science Principles** **Exam: Format of Assessment – 2 Hours – 60% of Exam Score**  **Multiple Choice | 74 Questions | 2 Hours**  **AP Computer Science Principles** **Through-Course Performance Tasks: 2 – 40% of Exam Score**  • Explore – Impact of Computing Innovations – 8 Hours (classroom time)  • Create – Application to Ideas – 12 Hours (classroom time)  **Link to DMPS Grading Resources:** <http://grading.dmschools.org>  **Link to Course Information @ AP Central:** <https://advancesinap.collegeboard.org/stem/computer-science-principles> |

|  |  |
| --- | --- |
| Grading Topics | College Board Curriculum Framework Alignment |
| Abstraction | Big Idea 2: Abstraction  EU2.1 A variety of abstractions built on binary sequences can be used to represent all digital data.  EU2.2 Multiple levels of abstraction are used to write programs or create other computational artifacts.  EU2.3 Models and simulations use abstraction to generate new understanding and knowledge. |
| Data and Information | Big Idea 3: Data and Information  EU3.1 People use computer programs to process information to gain insight and knowledge.  EU3.2 Computing facilitates exploration and the discovery of connections in information.  EU3.3 There are trade-offs when representing information as digital data. |
| Algorithms | Big Idea 4: Algorithms  EU4.1 Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented by programming languages.  EU4.2 Algorithms can solve many, but not all, computational problems. |
| Programming | Big Idea 5: Programming  EU5.1 Programs can be developed for creative expression to satisfy personal curiosity, new knowledge, or to solve problems to help people, organizations, or society.  EU5.2 People write programs to execute algorithms.  EU5.3 Programming is facilitated by appropriate abstractions.  EU5.4 Programs are developed, maintained, and used by people for different purposes.  EU5.5 Programming uses mathematical and logical concepts. |
| The Internet | Big Idea 6: The Internet  EU6.1 The Internet is a network of autonomous systems.  EU6.2 Characteristics of the Internet influence the systems built on it.  EU6.3 Cybersecurity is an important concern for the Internet and the systems built on it. |
| Creativity | Big Idea 1: Creativity  EU1.1 Creative development can be an essential process for creating computational artifacts.  EU1.2 Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.  EU1.3 Computing can extend traditional forms of human expression and experience. |
| Global Impact | Big Idea 7: Global Impact  EU7.1 Computing enhances communication, interaction, and cognition.  EU7.2 Computing enables innovation in nearly every field.  EU7.3 Computing has global effects – both beneficial and harmful – on people and society.  EU7.4 Computing innovations influence and are influenced by the economic, social, and cultural contexts in which they are designed and used.  EU7.5 An investigative process is aided by effective organization and selection of resources. |

**Standards-Referenced Grading Basics**

The teacher designs instructional activities and assessments that grow and measure a student’s skills in the elements identified on our topic scales. Each scale features many such skills and knowledges, also called learning targets. These are noted on the scale below with letters (A, B, C) and occur at Levels 2 and 3 of the scale. In the grade book, a specific learning activity could be marked as being 3A, meaning that the task measured the A item at Level 3.

|  |
| --- |
| **The Body of Evidence in a Process-Based Course** |
| **Process-Based SRG** *is defined as an SRG course design where the same scale recurs throughout the course, but the level of complexity of text and intricacy of task increase over time.*  AP Computer Science Principles does have a traditional unit-based design. In some topics, however, students cycle through the same topic repeatedly as they progress through the course, with changing content and an increasing complexity of the coding, analysis, and expectations throughout.  To account for this, process-based courses like this have their evidence considered in a “Sliding Window” approach. When determining the topic score for any given grading topic, *the most recent evidence* determines the topic score. Teacher discretion remains a vital part of this determination, but it is hard to overlook evidence from the most recent (and therefore rigorous) assessments. |





|  |
| --- |
| Abstraction |
| **Text and Resources** |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **4** | **3** | **2** |
| **Abstraction** | *In addition to meeting the learning goal, the student demonstrates in-depth inferences and applications that go beyond the goal.* | 3A: Explain how binary sequences are used to represent digital data.  3B: Represent phenomena and formulate, refine, and test hypotheses using models and simulations. | 2A: Identify the different types of binary sequences in an algorithm.  2B: Recognize components of an existing model or simulation for its content and algorithms.  2A/B: Identify and define vocabulary terms such as: Base Systems, Number Systems, Logic Gates, Constants, Expressions, Statements, Procedures, and Libraries, Boolean Operators. |

|  |
| --- |
| Data and Information |
| **Text and Resources** |
| Feedback: You are moving from recall/comprehension to a mix of analysis and knowledge utilization. Marzano (the word of Bob) wants you to go up like a set of stairs. If your 2s are in comprehension, then your 3s should be in analysis, and your 4s in knowledge utilization.  Consider: 3A: Test (ok) 3B: Evaluate (?) 3C: Determine (instead of analyze, which is a level of thinking, but a vague verb for learning targets). |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **4** | **3** | **2** |
| **Data and Information** | *In addition to meeting the learning goal, the student demonstrates in-depth inferences and applications that go beyond the goal.* | 3A: Test hypotheses about digitally processed information to gain insight and knowledge.  3B: Investigate how the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language.  3C: Analyze how data representation, storage security, and transmission of data involve computational manipulation of information. | 2A: Recognize data can processed to gain insight and knowledge.  2B: Recognize trends and connections from extracted data information.  2C: Describe the role large data sets play in computation processes.  2A/B/C: Identify and define vocabulary terms such as: Large data sets, Spreadsheets, Databases, Metadata, Sorting and Searching, Storage, Privacy |

|  |
| --- |
| Algorithms |
| **Text and Resources** |
| 3A: Ok  3B: Explore (?) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **4** | **3** | **2** |
| **Algorithms** | *In addition to meeting the learning goal, the student demonstrates in-depth inferences and applications that go beyond the goal.* | 3A: Develop an algorithm for implementation in a program.  3B: Investigate algorithms analytically and empirically for efficiency, correctness, and clarity. | 2A: Complete a missing component of a non-working or incomplete algorithm  2B: List and describe the ways an algorithm can be more efficient, correct, and clear  2A/B: Identify and define vocabulary such as: Programming languages (Sequential programming versus Object-oriented programming), Runtime, Heuristic approach |

|  |
| --- |
| Programming |
| **Text and Resources** |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **4** | **3** | **2** |
| **Programming** | *In addition to meeting the learning goal, the student demonstrates in-depth inferences and applications that go beyond the goal.* | 3A: Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge.  3B: Explain how programs implement algorithms.  3C: Use abstraction to manage complexity in programs.  3D: Employ appropriate mathematical logical concepts in programming. | 2A: Draft a program for creative expression, to satisfy personal curiosity, or to create new knowledge.  2B: Recognize how programs implement algorithms.  2C: Describe how abstractions increase efficiency and manage complexity.  2D: List mathematical logical concepts used in programming.  2A/B/C/D: Identify and define vocabulary such as: Iterative process, Program documentation, debug, CPU, Input and Output, Procedures, Variables, Data Abstraction, Strings, Concatenations, Application program interfaces (API), Abstract data types |

|  |
| --- |
| The Internet |
| **Text and Resources** |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **4** | **3** | **2** |
| **The Internet** | *In addition to meeting the learning goal, the student demonstrates in-depth inferences and applications that go beyond the goal.* | 3A: Explain the abstractions, hierarchy, and the characteristics of the internet and how it functions.  3B: Given existing cybersecurity concerns research and critique potential options to address these issues with the Internet and systems built on it. | 2A: List the components of the internet and the role the play.  2B: Identify existing cybersecurity concerns.  Identify and define vocabulary terms such as: DNS, IP addressing, HTTP, SMTP, IETF, DDoS, phishing, virus, firewall, cryptography, encryption, public key security, certificate authority (CA), redundancy |

|  |
| --- |
| Creativity |
| **Text and Resources** |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **4** | **3** | **2** |
| **Creativity** | *In addition to meeting the learning goal, the student demonstrates in-depth inferences and applications that go beyond the goal.* | 3A.1: Create a computational artifact for creative expression.  3A.2: Analyze the correctness, usability, functionality of computational artifacts.  3A.3: Use computing tools and techniques for creative expression.  3B: Collaborate in the creation of computational artifacts. | 2A: Draft a computational artifact for creative expression.  2B: List ways one can be collaborative in the creation of a computational artifacts.  Identify and define vocabulary terms such as: Artifact, Integrated Development Environment (IDE) |

|  |
| --- |
| Global Impact |
| **Text and Resources** |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **4** | **3** | **2** |
| **Global Impact** | *In addition to meeting the learning goal, the student demonstrates in-depth inferences and applications that go beyond the goal.* | 3A.1: Explain how computing innovations affect communication, interaction, and cognition.  3A.2: Explain how computing has impacted innovations in other fields.  3B: Analyze the beneficial and harmful effects of computing. | 2A: Recognize computing innovations affect communication, interaction, and cognition in many fields including economic, social, and cultural contexts.  2B: Recognize and list beneficial and harmful effects of computing.  Identify and define vocabulary terms such as: Email, SMS, The Cloud, GPS, Smart Technologies, Moore’s Law, Digital Millennium Copyright Act |