Curricular Requirements

CR1	The teacher and students have access to a college-level computer science textbook, in print or electronic format.	
CR2	The course provides opportunities to develop student understanding of the required content outlined in each of the big ideas described in the AP Course and Exam Description.	
CR3	The course provides opportunities to develop student understanding of the big ideas, as outlined in AP Course and Exam Description (CED).	
CR4	The course provides opportunities for students to develop the skills related to the Computational Thinking Practice 1: Solution Design, as outlined in AP® Course and Exam Description.	
CR5	The course provides opportunities for students to develop the skills related Computational Thinking Practice 2: Algorithms and Program Development, as outlined in AP Course Exam Description	
CR6	The course provides opportunities for students to develop the skills related Computational Thinking Practice 3: Abstraction in Program Development, as outlined in the AP Course and Exam Description.	
CR7	The course provides opportunities for students to develop the skills related Computational Thinking Practice 4: Code Analysis, as outlined in AP Course and Exam Description.	
CR8	The course provides opportunities for students to develop the skills related Computational Thinking Practice 5: Computing Innovations, as outlined in AP Course and Exam Description.	
CR9	The course provides opportunities for students to develop the skills related Computational Thinking Practice 6: Responsible Computing, as outlined in AP Course and Exam Description.	
CR10	The course provides a minimum of 3 opportunities for students to investigate different computing innovations.	
CR11	Students are provided at least 12 hours of dedicated class time to complete the Create Performance Task.	

Explore Curricular Requirements

In this course, students will address the impacts of computing through projects in units 1, 2, and 10. In unit 10, students will more deeply dive into a particular computational innovation and will answer the following prompts.

- Explain beneficial and harmful effects a computing innovation has on society, economy, or culture.
- Identify the data used by the computing innovation and explain how the data is consumed, produced, or transformed by the given computing innovation.
- Identify data privacy, security, or storage concerns for the computing innovation.

Course Description

AP® Computer Science Principles introduces students to the breadth of the field of computer science. This course is equivalent to a first-semester, college-level breadth course in computer science. AP® Computer Science Principles is designed to encourage a diverse group of students to explore computer science.

This course introduces students to a broad set of big ideas. These big ideas are creative development, data, algorithms and programming, computing systems and networks, and the impact of computing.

Additionally, this course emphasizes the use of computational thinking practices for effective learning experiences and problem solving. These practices include computational solution design, algorithms and program development, abstraction in program development, code analysis, computing innovations, and responsible computing. In this course, students will learn to design and evaluate solutions and to apply computer science to solve problems through the development of algorithms and programs. They will incorporate abstraction into programs and use data to discover new knowledge. Students will also explain how computing innovations and computing systems, including the Internet, work, explore their potential impacts, and contribute to a computing culture that is collaborative and ethical.

This course introduces students to a broad set of big ideas. These big ideas are;

- Creative Development (CRD)
- Data (DAT)
- Algorithms and Programming (AAP)
- Computing Systems and Networks (CSN)
- The Impact of Computing (IOC)

In addition, this course emphasizes the use of computational thinking practices for effective learning experiences and problem solving. These practices include:

- Computational Solution Design (CTP1)
- Algorithms and Program Development (CTP2)
- Abstraction in Program Development(CTP3)
- Code Analysis (CTP4)
- Computing Innovations (CTP5)

- Responsible Computing. (CTP6)

In this course, students will learn to design and evaluate solutions and to apply computer science to solve problems through the development of algorithms and programs. They will incorporate abstraction into programs and use data to discover new knowledge. Students will also explain how computing innovations and computing systems, including the Internet, work, explore their potential impacts, and contribute to a computing culture that is collaborative and ethical.

Programming Language and Environments

The coding language chosen for this course is Python. Students will also be introduced to block-based coding environment with Scratch for several activities.

Resources [CR1]

- Parsons, June Jamrich. (2018). New Perspectives on Computer Concepts 2018: Comprehensive | 20th Edition. Cengage Learning, Inc.
- Open Source. CS Principles: Big Ideas in Programming. Runestone Academy
- Various websites that provide videos and activities relevant to computer science
- http://apcentral.collegeboard.com (Teachers and students are encouraged to make use of the additional resources provided through the website. Students may access additional instructional supports, complete progress checks, and take practice exams.)

Course at a Glance [CR2]

Unit 1- The Digital World	What are computers for? How do they operate on data at the lowest level – bits and bytes? How do programs tell the computer what to do? Students explore binary data and how it can be used to create things we know and enjoy.
Unit 2: The Internet	In this unit students learn about networks – what they are, how they work, how data travels within them. Students then explore the Internet – the largest and best-known network in the world – and its close cousin the World Wide Web.
Unit 3: Programming	In this unit, students are introduced to basic programming concepts and see how simple programs are organized.
Unit 4: Controlling Your Program Flow	In this unit students are introduced to programming techniques which allow them to control the flow of the program
Unit 5: Doing More with Algorithms	This unit takes students further still on the programming journey. They examine algorithms, manipulate lists, and make procedure calls.

Unit 6: Semester Review and Practice Exam	In this unit, students will review previous 5 units and take at least 1 AP CSP practice exam.
Unit 7: Modularity, Modeling, and Data Containers	In this unit students learn some more advanced programming concepts and techniques. They use pre-built libraries of functionality that save them from building those functions themselves. They create simulations.
Unit 8: Data	In this unit students learn how to work with data and how to identify and solve problems with their data. They also learn about transforming their data as needed, as well as maintaining good data hygiene.
Unit 9: Data Risks	Data is not necessarily benign. The use of computers and data entails many risks in the realms of authentication and identity as well as privacy and cyber security. In this unit students are exposed to all these issues.
Unit 10: Safety and Ethical Concerns	This unit covers intellectual property rights, the digital divide, bias in algorithms that have become a staple of decision-making, and the widespread presence of hackers.
Unit 11: Computers Changing the World	This unit concludes the course by looking at computers from the broadest perspective: how they are changing our world. Students discuss astonishing ways in which computing has led to new knowledge, medical breakthroughs, and open-source behavior.
Unit 12: AP Create Performance Task	Students will write a program that meets the requirements of the AP CSP Create Performance Task. Students will create a video demonstrating their program displaying the required features. Students will provide written responses to questions in the AP CSP Create Performance Task.

Course Outline [CR2]

Unit 1 – The Digital World

The course begins at the beginning: What are computers for? How do they operate on data at the lowest level – bits and bytes? How do programs tell the computer what to do? Students explore binary data and how it can be used to create things we know and enjoy, such as images and music.

Big Ideas: CRD, IOC, DAT, **[CR3]**, **[CR8]**

Topics for Overview

- How Have Computers Changed Your Life?
- What Language Do Computers Talk?
- Strings You Cannot Tie
- When Numbers Do Not Behave
- Compare Generations
- When Data is Music to Your Ears

- Breaking Images into Bits
- Data Compression
- Programming without Words

Activities

Interwoven in the lessons are short activities, such as playing the AI game Quick Draw, practicing binary to decimal conversions, and learning to use Scratch. The unit also has two formal projects:

- ***(Computing Innovation Project #1)*** Students investigate how computer innovations have changed the life of an adult over 50 years old. They do this by interviewing an adult over 50 years old about five computer innovations that have changed their life the most and in what ways. Students meet as a group and relate how each of the innovations relates to one of the five big ideas. **[CR 10]**
- Students analyze an image by creating a bit representation of a black and white image.

Unit 2 – The Internet

A computer in isolation has some usefulness, but it is limited. When connected to a network, that same computer has far greater value. In this unit students will explore the impacts the internet has had on society, including beneficial and harmful effects.

Big Ideas: CSN, IOC, **[CR3]**

Topics for Overview

- Network Building Blocks
- Building a Network
- Network Protocols
- Where's My Web?
- Is the Internet Worth Keeping?
- Fault Tolerance
- How the Web Grows
- Parallel and Distributed Computing
- What Has Distributed Computing Done for You?

Activities

Interwoven in the lessons are short activities, such as playing the AI game Quick Draw, practicing binary to decimal conversions, and learning to use Scratch. The unit also has two formal projects:

- ***(Computing Innovation Project #2)*** Students evaluate how the internet has impacted society, both positively and negatively, citing specific examples – whether it has been an overall benefit to society using at least three criteria determined by the student. **[CR 10]**
    - Impact of social media
    - Ease of accessing personal data
    - Online marketplaces
- Students compare the efficiency of performing a task sequentially, in parallel, or using the distributed model. They describe the benefits and challenges of parallel and distributed computing. For example, students could perform a physical task, such as folding 50 paper airplanes, using the three models. They will compare the efficiency of each model. One of the measures used should include the speedup. **[CR 8]**

Unit 3 – Programming

A computer is a machine designed to carry out any task it is asked to do if the task can be translated into instructions in a symbolic language. Software programs are sets of instructions to make the computer do such tasks. In this unit, students are introduced to basic programming concepts and see how simple programs are organized.

Big Ideas: AAP, CRD, DAT, **[CR3]**

Topics for Overview

- Computer Arithmetic
- Writing a Program
- What Happens When You Run a Program
- Planning Your Program
- Building Your Application
- Lists, the Everything Container
- Manipulating Lists
- Data Abstractions
- Getting Input
- Event-Driven Programs
- Design as an Iterative Process
- Design as a Collaborative Process

Activities **[CR 4]**

The unit has two activities:

- Students plan and write a program to perform at least 10 calculations.
- Students practice collaborating on a software project. As a team, they develop a plan for a program with a user interface as part of the plan.

Unit 4 – Controlling Your Program Flow

A sophisticated computer program is more than just a flat set of simple statements. At times it will need to perform simple and complex mathematical operations, make choices whether to go this way or that way, repeat a set of statements more than once, and handle errors. In this unit students are introduced to all these programming techniques.

Big Ideas: AAP, CRD, **[CR3]**

Topics for Overview

- When Errors Happen
- Testing Programs
- Manipulate Strings
- Relational Operators
- Logical Operators
- Conditionals
- Nested Conditionals
- Iteration
- Nested Loops
- While Loops

Activities [CR 5]

The unit has two activities:

- Students plan a project that uses at least two choices. They also plan how they will test for errors.
- Students write a program that uses at least two choices, consistent with their plan. Then they test for errors and correct any errors found.

Unit 5 – Doing More with Algorithms

This unit takes students further still on the programming journey. They examine algorithms, manipulate lists, and make procedure calls.

Big Ideas: AAP, CRD, **[CR5]**

Topics for Overview

- Comparing Algorithms
- Modifying Existing Algorithms
- Manipulating Lists
- Linear Searching
- Binary Search
- Calling Procedures
- Interpreting the Exam Procedure Call
- How Functions /Procedures Manage Complexity
- What's My Procedure?
- Sending Output

Activities [CR 6]

The unit has two activities:

- Students will manipulate lists in a program. They write a program that manipulates lists in at least three ways, including a search.
- Students write a program with two functions that process lists.

Unit 6 – Semester Review and Exam

In this unit, students will review the previous 5 units. **[CR1]**

Unit 7 – Modularity, Modeling, and Data Containers

In this unit students learn some more advanced programming concepts and techniques. They use pre-built libraries of functionality that save them from building those functions themselves. They create

simulations. And they step back to examine what kinds of problems computers would be good – or not good – at solving.

Big Ideas: AAP, DAT, **[CR3]**

Topics for Overview

- Libraries
- Pseudorandom Numbers
- The Secret Module
- Simulations
- Bias in Simulations
- Using a Two-Dimensional List to Extract Data
- Dictionaries
- The Heuristic Approach
- Is Your Algorithm Efficient?
- Problems Computers Cannot Solve

Activities

Interwoven in the lessons are simulations that model real-world phenomena. The unit also has two formal projects:

- Students describe a real-world simulation and document their source in their report.
- Students write a program that uses a pseudorandom number generator in a loop to simulate a choice and report their results. **[CR 5]**

Unit 8 – Data

A computer program without data is like a racing car without a track; it has nowhere to go and nothing to do. Data is the life blood of computer software. In this unit students learn how to work with data and how to identify and solve problems with their data. They also learn about transforming their data as needed, as well as maintaining good data hygiene.

Big Ideas: DAT, **[CR3]**

Topics for Overview

- What Can Data Tell You?
- Data about Data
- Processing Data
- Problems with Data
- Solutions to Data Problems
- Data Tools
- Searching for Data
- Spreadsheets
- Data Transformations
- Data Cleaning Benefits

Activities

The unit has two activities:

- Students engage in a data collection exercise. They ask a question and collect data to answer the question. They will analyze their data in the next project.

- Students write a program using a list and a function to analyze the data collected in the previous project. This project helps prepare them for the CPT.

Unit 9 – Data Risks

Data is not necessarily benign. The use of computers and data entails many risks in the realms of authentication and identity as well as privacy and cyber security. In this unit students are exposed to all these issues.

Big Ideas: IOC, DAT, AAP, CRD, **[CR3]**

Topics for Overview

- Risks to Private Information
- Authentication Measures
- Computer Viruses
- Making Web Pages Safe
- Who Is Listening?
- Bits with Unusual Consequences
- A Duck Might Not Be a Duck
- Mistaken Identity
- Potential List Problem
- Debugging

Activities **[CR 9]**

The unit has two activities:

- Students explore privacy issues by writing a plan to protect their personally identifiable information (PII).
- Students research a recent cyberattack. They then describe the attack, its consequences, and how it was resolved.

Unit 10 – Safety and Ethical Concerns

Computer technology has changed our daily lives and our world at large in countless ways. With the benefits have come a variety of social problems that students explore in this unit: intellectual property rights, the digital divide, bias in algorithms that have become a staple of decision-making, and the widespread presence of hackers.

Big Ideas: IOC, CRD, AAP, **[CR3]**

Topics for Overview

- Can Computing Innovations Be Harmful?
- Accessibility in the World Wide Web
- Bias in Computing Algorithms
- Hackers
- Intellectual Property Rights
- Code Sharing
- Digital Divide
- Crossing the Digital Divide
- When Collaborations Fail

Activities

The unit has two activities:

- ***(Computing Innovation Project #3)*** **Explore Curricular Requirement:** Students will address security concerns in a computer innovation. They will explain beneficial and harmful effects of the computing innovation on society, economy, or culture, identify the data used in the computing innovation and explain how the data is consumed, produced, or transformed by the given computing innovation. Students will also identify data privacy, security, or storage concerns for the computing innovation. They will present their information and acknowledge their sources of information. **[CR10]**
- Each student performs a code review of another student's project. **[CR7]**

Unit 11 – Computers Changing the World

This unit concludes the course by looking at computers from the broadest perspective: how they are changing our world. Students discuss relevant ways in which computing has led to new knowledge, medical breakthroughs, and open-source behavior.

Big Ideas: IOC, DAT, **[CR3]**

Topics for Overview

- The Internet is Not What it Was Planned to Be
- Surprises in the Computing World
- Open Source and Other Amazing Examples of Generosity
- Crowdsourcing
- APIs
- Plan Your Performance Task
- Combining Data
- Data Transformations
- Medical Data Breakthroughs
- Are You Ready for the CTP?

Activities **[CR 4]**

The unit has two activities:

- Students plan a Create Performance Task (CPT).
- Students present one of their previous programs to an audience and record a video, demonstrating the purpose, the input, and the output.

Unit 12 – AP Create Performance Task and Semester Exam [CR11]

Students will write a program that meets the requirements of the AP CSP Create Performance Task. Students will create a video demonstrating their program displaying the required features. Students will provide written responses to questions in the AP CSP Create Performance Task.

In this unit, students will review the previous 5 units and take the semester exam.

Unit 13 – Final Exam and AP Practice Exam

In this unit, students will review the course and take the final exam.

The student will take an AP Practice Exam.