AP Computer Science Principles Course Description

This workshop provides teachers with the tools they need to implement an effective AP Computer Science Principles course. During this training, teachers will explore the computational thinking practices and the components of the curriculum framework, including the big ideas, enduring understandings, learning objectives, and essential knowledge. Participants will understand how to use activities that organize the course content to develop students' proficiencies in the skills identified by the curriculum framework. In addition, participants will work on a course plan that will help them decide how they will teach the skills and content of the AP Computer Science Principles course.

AP Computer Science Principles
4-5 Day Workshop Agenda

(30 hours)

	Agenda -Day 1
	Understanding the Course
45 min	Lesson 1: AP Computer Science Principles: Engaging All Students
1 hour	Lesson 2: Computational Thinking Practices and the AP Computer Science Principles course
1.5 hours	Lesson 3: Developing Student Understanding
2.5 hours	Lesson 4: Understanding the Learning Objectives
1.5 hour	Lesson 5: Understanding the Big Ideas
	Agenda -Day 2
1 hour	Lesson 17: Assessment Part 1: Explore Performance Task
	Planning Your Course
3 hours	Lesson 6: Planning Your Course
	Computational Thinking Practices
45 min	Lesson 7: Connecting Computing
45 min	Lesson 8: Creating Computational Artifacts
2 hours	Lesson 9: Abstracting
	Agenda -Day 3
1 hour	Lesson 17: Assessment Part 2: Create Performance Task
2 hours	Lesson 10: Analyzing Problems and Artifacts
1.5 hours	Lesson 11: Communicating
1.5 hours	Lesson 12: Collaborating
	Teaching the AP Computer Science Principles Course
1 hour	Lesson 13: Sequencing the AP Computer Science Principles Course
	Agenda -Day 4
1 hour	Lesson 14: Selecting Resources to Support Teaching AP Computer Science Principles
2 hours	Lesson 15: Strategies for Teaching AP Computer Science Principles
2 hours	Lesson 16: Unit Development
1 hour	Lesson 17: Assessment Part 3: Resources and Strategies
	Curricular Requirements and Syllabus Development
1 hour	Lesson 18: Curricular Requirements and Syllabus Development

Teacher Understandings for AP Computer Science Principles multi-day workshops

Key Takeaways (what the participants discover) *Teachers will understand that…*	**Learning Goals** (what the participants do) *Teachers will be able to…*	**Knowledge** (the information that gets taught) *Teachers will know…*
Lesson 1		
An underlying goal of AP Computer Science Principles is to broaden participation.	Explain the goals of computer science as it relates to STEM education and broadening participation.	The estimated need for CS educated citizens based on the projections from the Bureau of Labor and Statistics.
		The field of CS is lacking in female and minority groups.
	Develop strategies to make AP Computer Science Principles engaging for all students.	AP Computer Science Principles is meant to be engaging and accessible to all students.
		Instructional strategies to engage students in innovations while discovering their interests and backgrounds.
Lesson 2		
In order to be successful on the AP Computer Science Principles Exam, students will need to demonstrate proficiency with the computational thinking practices.	Recognize computational thinking practices when examining assessment questions from the AP Computer Science Principles assessments.	The six computational thinking practices students will be expected to apply for the AP Computer Science Principles course.
Lesson 3		
By examining Performance Task, student samples and rubrics, teachers can gauge the proficiency level with the enduring understandings students will be required to demonstrate.	Identify what computational thinking practices and enduring understandings are being demonstrated in student samples and which need to be developed further by the student.	The various levels of skill demonstrated on the through course assessment.
		New understandings emerge when students build on prior knowledge.
Lesson 4		

Key Takeaways (what the participants discover) *Teachers will understand that…*	**Learning Goals** (what the participants do) *Teachers will be able to…*	**Knowledge** (the information that gets taught) *Teachers will know…*
Knowledge of the content is essential for building understanding and requires teachers to constantly return to the curriculum framework in order to understand the scope of the course content and what exclusions may exist.	Connect the enduring understandings, learning objectives and essential knowledge statements to the computational thinking practices and identify ways to scaffold and combine them to create a cohesive curriculum.	The relationship between the enduring understandings, learning objectives and essential knowledge and how they define the scope of the course.
Lesson 5 The big ideas do not necessarily represent individual units to be taught and often times, multiple big ideas can be addressed in the same unit or lesson.	Create activities which include more than one big idea.	The seven big ideas in AP Computer Science Principles.
Lesson 6 Success in teaching this course is reliant on understanding who your students are and designing a course that meets their instructional needs and matches their interests.	Identify different course approaches used in the course planning and pacing guides for AP Computer Science Principles.	A project-based approach involves presenting students with an outline for a long-term project or problem to solve.
		An integrated approach spirals the curriculum to address multiple big ideas in the same project or unit.
		An inquiry-based approach allows students' questions to guide instruction.
	Create a high level, year-long plan for teaching AP Computer Science Principles that takes into consideration the students who will be taking the class.	Different instructional approaches to teaching AP Computer Science Principles.
		The Explore Performance Task will require 8 hours of class time, while the Create Performance Task will require 12 hours of class time.

Key Takeaways (what the participants discover) *Teachers will understand that…*	**Learning Goals** (what the participants do) *Teachers will be able to…*	**Knowledge** (the information that gets taught) *Teachers will know…*
	Identify areas of improvement and adjust teaching based on feedback from Instructional Planning Reports.	How feedback is represented in the charts and tables of the Instructional Planning Report.
		The categories used to represent the learning objectives and which learning objectives are grouped under each category.
	Apply recruitment strategies to form AP Computer Science Principles classes that are representative of their school population.	Effective recruitment strategies to recruit more female students into computer science.
		Effective recruitment strategies to recruit underrepresented minorities.
Lesson 7 Connections between people and computing can be seen in the innovations being developed. Innovations can be used to illustrate the connection between computing and multiple big ideas.	Create an activity integrating a computational thinking practice and a big idea.	Instructional strategies appropriate to scaffold students' learning to connect computing.
Students should be able to identify the implications computers have on individuals, society, commercial markets, and innovations.	Identify beneficial and harmful effects of computing innovations.	Innovations enabled by computing raise legal and ethical concerns.
Lesson 8 The skills required to create computational artifacts need to be scaffolded and taught. Teachers should develop incremental performance tasks to build student skills instead of assigning the tasks in their entirety.	Incorporate effective instructional strategies into activities to scaffold students' ability to create computational artifacts.	Instructional strategies specifically helpful in scaffolding the learning of skills required to create computational artifacts.
		Skills need to be incrementally taught prior to being assigned.

Key Takeaways (what the participants discover) *Teachers will understand that…*	**Learning Goals** (what the participants do) *Teachers will be able to…*	**Knowledge** (the information that gets taught) *Teachers will know…*
	Apply a development process to the creation of computational artifacts.	Development processes, such as the engineering design process or the software development life cycle, provide students with a way of organizing the design of their artifacts around planning, reflection and revision.
Students should be able to design and develop interesting computing artifacts as well as apply computing techniques to creatively solve problems.	Develop activities for students to design and develop relevant computational artifacts.	Examples of effective and ineffective computational artifacts.
		A variety of effective computational tools that students could use to create computational artifacts.
Lesson 9 Abstracting allows us to use generalizations to make our programs more reusable and understandable. Breaking programs into subtasks and creating procedures around these subtasks is one way that we create abstraction.	Break programs into subtasks and develop abstractions through the creation of procedures.	Software is developed using multiple levels of abstractions, such as constants, expressions, statements, variables, procedures, and parameters.
Abstraction is used in models and simulations to formulate, refine and test hypotheses.	Provide their students with real-world examples of models and simulations.	How abstracting is used in modeling in AP Computer Science Principles.
		Simulations mimic real-world events without the cost or danger of building and testing the phenomena in the real world.
The computer is able to make meaning out of bits to create text, pictures, colors, etc.	Apply knowledge of how a computer stores data as bits to how a computer stores other larger collections of data such as pictures, colors, videos, email, etc.	At the lowest level, all digital data are represented as bits.

Key Takeaways (what the participants discover) *Teachers will understand that…*	**Learning Goals** (what the participants do) *Teachers will be able to…*	**Knowledge** (the information that gets taught) *Teachers will know…*
		Number bases, including binary, are used to represent digital data.
		At one of the lowest levels of abstraction, digital data is represented in binary (base 2) using only combinations of the digits zero and one.
		A fixed number of bits used to represent integers limits the range of integer values; this limitation can result in overflow or other errors.
		Numbers can be converted from any base to any other base.
Lesson 10 The development of algorithms allows students to break down and solve problems in a variety of ways. There are multiple solutions to a problem each with their own benefits.	Apply different algorithm structures to the solution of a problem.	Different ways to plan and design computational artifacts and solutions.
		One strategy to help students manage the complexity and solve problem is to chunk the problem into smaller more manageable pieces called procedures.
		Flowcharts and pseudocode are useful ways to represent algorithms. Each has its own best time to be used.
Developing test cases as part of planning a computational program artifact assists in determining the correctness of the program.	Develop test cases which can be used to determine the correctness, accuracy and function of a computational artifact.	When using Boolean operators for decisions, you will want to test your program in all situations.

Key Takeaways (what the participants discover) *Teachers will understand that…*	**Learning Goals** (what the participants do) *Teachers will be able to…*	**Knowledge** (the information that gets taught) *Teachers will know…*
	Locate and correct errors in programming code.	Tracing is an effective strategy used to locate logic errors.
		Collaboration and the use of pair programming is an effective way to locate and correct errors in programming code.
A variety of programming languages and platforms can be used to solve the same problem.	Apply knowledge of programming structure to create a program that is used to solve a problem, for creative expression or provide the viewer with new insight or knowledge.	A variety of resources are available to help learn different programming platforms.
The different ways that data can be represented, modelled and analyzed to answer questions and gain insight and knowledge.	Draw a conclusion based on different data representations.	Spreadsheets can be used to create different representations of data that can be helpful to gain insight and knowledge.
Lesson 11 Effective communication, whether it is verbal or written, requires modeling and practice for students to master.	Communicate the solution to a problem verbally and in written form.	The requirements of technical writing and communication are different from writing students do for other class.
		Writing in computer science needs to be modeled and taught. Students will need adequate feedback on any writing practice.
		Instructional strategies that are effective for teaching students technical writing.
	Recognize computational artifacts as visuals that can communicate information and ideas.	Visual representations and computational artifacts can be effective forms of communication.

Key Takeaways (what the participants discover) *Teachers will understand that…*	**Learning Goals** (what the participants do) *Teachers will be able to…*	**Knowledge** (the information that gets taught) *Teachers will know…*
Algorithms are used to develop and express solutions to computational problems.	Draw a flowchart to represent a procedure.	Symbols used in the creation of a flowchart.
Using relevant and credible sources is a way to engage students while building the skill of connecting computing.	Locate relevant and credible sources related to computing innovations.	The importance of using relevant and interesting sources to build the skill of describing the connection between people and computing as well as understanding of the big ideas.
		Using relevant and credible sources builds the students' credibility when answering the prompts.
	Evaluate sources for credibility and relevance.	All resources should be evaluated for credibility and relevance.
		A source found on the Internet may not be credible.
		A source may be related, but not relevant to the investigation.
Lesson 12 Collaboration comes in many forms. Students should have many opportunities to practice collaboration in a variety of situations and ways.	Facilitate students' collaboration with peers and resolve communication problems while students are creating computational artifacts.	Instructional strategies to use to foster collaboration in the classroom.
		The six ways students are expected to collaborate as outlined in the curriculum framework.
Lesson 13 The computational thinking practices need to be broken down and taught, starting at the students' entry level through mastery.	Recognize the ability level of students as an indicator of where to begin teaching the computational thinking practices.	The level in which you will need to start teaching the computational thinking practices depends on the students in your classroom.

Key Takeaways (what the participants discover) *Teachers will understand that…*	**Learning Goals** (what the participants do) *Teachers will be able to…*	**Knowledge** (the information that gets taught) *Teachers will know…*
	Break down the computational thinking practices to understand the incremental challenges that will be scaffolded to move students towards mastery.	Skills development must be sequenced and scaffolded so that students are able to build on prior knowledge to develop their understanding.
Lesson 14 Resources should be selected based on their alignment with the AP Computer Science Principles Curriculum Framework and meet the needs of their students.	Select quality resources to use with students that reinforce computational thinking practices, the big ideas and reflect the current changes in the field of computer science.	ACM Tech News is a reliable resource for finding current articles on computing innovations.
		A list of aligned textbooks can be found through course audit.
The background of your students should influence decisions such as choice of programming language.	Select a programming language or environment that suit their students' abilities and interests.	AP Computer Science Principles does not specify a language in which to teach the programming big idea. Teachers should choose a language that is most suitable for their students.
		Teachers can choose either block-based or text-based programming languages, or a combination of both.
		The availability and location of resources that can be used to teach programming to students.
		Block-based programming languages are sometimes easier for students who have little or no programming experience.
Lesson 15		

Key Takeaways (what the participants discover) *Teachers will understand that…*	**Learning Goals** (what the participants do) *Teachers will be able to…*	**Knowledge** (the information that gets taught) *Teachers will know…*
Cooperative Learning strategies require students to use collaboration and communication skills which are essential to solving computational problems and creating computational artifacts.	Design instruction that is engaging to all students, in particular underrepresented minorities and female students.	Instructional strategies that can be used to make content more meaningful and accessible to students.
Lesson 16 Assessment, instruction, and learning goals should be aligned.	Build units of instruction that align assessment with learning goals and instruction.	The Understanding by Design approach to unit development.
Lesson 17 Assessments should be used so to assess skills and content students have been taught. Feedback is a key component of effective assessments.	Use assessments to check for student understanding.	The performance qualities for both the Create and Explore Performance Tasks.
	Create incremental activities and assessment that are meant to support the building of understandings.	The computational thinking practices and enduring understanding required for students to perform at the highest level need to be scaffolded and taught.
		Practicing the entire performance task will not teach students the skills required to be successful.
	Provide students with feedback on incremental activities and assessments that are meant to support the building of understandings.	Effective strategies to use to scaffold skills and move students from the lowest level to the middle level to the highest level of performance.
Lesson 18 The curricular requirements ensure that all AP Computer Science Principles teachers are teaching a college-level course.	Identify curricular requirements that are met in sample syllabi.	The curricular and resource requirements for AP Computer Science Principles.

Key Takeaways (what the participants discover) *Teachers will understand that…*	**Learning Goals** (what the participants do) *Teachers will be able to…*	**Knowledge** (the information that gets taught) *Teachers will know…*
	Design a course syllabus that meets all curricular requirements.	The level of specificity required for meeting the curricular requirements in a syllabus.