FAIRFIELD PUBLIC SCHOOLS

Advanced Placement
Computer Science - Principles

Course Information

Grade(s):	9-12
Discipline/Course:	Business
Course Title:	AP Computer Science - Principles
Prerequisite(s):	Algebra 1 (B or Better) It is recommended that students in the AP Computer Science Principles course have successfully completed a first-year high school algebra course with a strong foundation of basic linear functions, the composition of functions, and problem-solving strategies that require multiple approaches and collaborative efforts. In addition, students should be able to use a Cartesian (x, y) coordinate system to represent points on a plane. It is important that students and their advisers understand that any significant computer science course builds upon a foundation of mathematical reasoning that should be acquired before attempting such a course. Prior computer science experience is not required to take this course.
Course Description: *Program of Studies*	AP Computer Science Principles offers a multidisciplinary approach to learning the underlying principles of computation. The course will introduce students to the creative aspects of programming, abstractions, algorithms, large data sets, the Internet, cybersecurity concerns, and computing impacts. Computer Science Principles will give students the opportunity to use technology to address real-world problems and build relevant solutions. Together, these aspects of the course make up a rigorous and rich curriculum that aims to broaden participation in computer science.
Course Essential Questions:	● Is coding a creative pursuit? How? ● How are algorithms implemented and executed on computers and computational devices? ● How are programs used for creative expression, to satisfy personal curiosity or to create new knowledge? ● How do computer programs implement algorithms? Are there problems that can't be solved by algorithms?
Course Enduring	Big Ideas 1-5

Understandings:	The big ideas serve as the foundation of the course and help students create meaningful connections among concepts. They are often overarching concepts or themes that become threads that run throughout the course. Revisiting the big ideas and applying them in a variety of contexts enables students to develop deeper conceptual understanding. Below are the big ideas of the course and a brief description of each. Big Ideas Exam Weighting Big Idea 1: Creative Development 10–13% Big Idea 2: Data 17–22% Big Idea 3: Algorithms and Programming 30–35% Big Idea 4: Computer Systems and Networks 11–15% Big Idea 5: Impact of Computing 21–26%
Duration and credit:	2 Semesters (1 year) / 1.0 credits
Course Materials/Resources:	AP Computer Science Principles Course and Exam Description, Effective Fall 2020 (collegeboard.org) Choosing a Programming Language AP Computer Science Principles does not require the use of a specific programming language. Because a goal of this course is to broaden participation, teachers can center their course around computing concepts in the course framework that support the creation of exciting and relevant computational artifacts. For this reason, teachers are encouraged to select the programming language that is most appropriate for their classroom and that will provide students opportunities to successfully engage with the course content. The programming language selected should contain functionality that is specified in the course framework and performance tasks. Appropriate programming languages for this course are ones that allow students to evaluate expressions, develop procedures, and use variables, lists, conditionals, and loops. The following is a noncomprehensive list of programming languages or development environments that can be considered for use in this course: (REF: College Board, AP Computer Science Principles Course and Exam Description, 2020) ● Alice This block-based programming language includes a 3-D modeling environment that allows students to create and animate 3-D worlds. This environment lends itself well to creating stories and games.

	● App Inventor This open-source web application is block-based and allows students to create their own applications on mobile devices. ● App Lab This is a programming environment for creating web applications with JavaScript. It allows students to develop programs and toggle back and forth between blockbased and text-based programming modes. ● EarSketch This text-based browser application allows students to create their own music using either JavaScript or Python. ● Greenfoot This text-based Java IDE is designed for use in education to create 2-D graphic applications, such as simulations and interactive games. ● Java This text-based programming language allows students to create and solve problems that vary widely in difficulty. There are several IDEs that can be used to write programs in Java. ● JavaScript This text-based programming language is commonly used to create interactive effects within web browsers. ● LEGO Mindstorms EV3 This product integrates block-based programming with LEGO bricks and sensors to create and program robots. The instructions are assembled by linking together function blocks. ● Microsoft MakeCode This development environment provides both block and text editors for students at different levels. Microsoft MakeCode is a free, open-source, web-based environment with open educational resources for teachers. ● Processing - This text-based programming language was initially created to serve as a software sketchbook, and it can be used to teach programming in a visual context. ● Python This text-based programming language has the benefit of readability, which might be helpful to new programmers. ● Quorum This universally designed and evidence-based programming language allows students, including those with disabilities, to create programs for gaming, robotics, sound processing, networking, and more. ● Scratch This block-based programming language allows students to build scripts to run animations. This product can be downloaded and installed on a computer or run in a browser. ● Snap! Snap! combines the power of text-based languages such as Python or JavaScript with the visual simplicity of block-based Scratch. It can grow with your students, because new blocks can be written in Snap! itself or in JavaScript.

	• Swift This powerful and intuitive programming language can be used for macOS, iOS, watchOS, tvOS and beyond. Students gain practical experience with the tools and techniques to build basic iOS apps with Swift and Xcode, an IDE at the center of the Apple development experience. • TI-Basic This is an introductory text-based programming language for TI graphing calculators. Programs with the TI-Innovator Hub can read inputs from sensors and send output to control speakers, LEDs, motors, a robotic vehicle, and more. AP Computer Science Principles Explore essential resources for AP Computer Science Principles. https://apcentral.collegeboard.org/courses/ap-computer-science-principles
FPS Course Academic Expectation(s):	Synthesizing and Evaluating The student weighs evidence, arguments, claims and beliefs in order to critically and effectively solve problems and to justify conclusions**.** Creating and Constructing The student transforms existing ideas and knowledge into original ideas, products, and processes.
Year at a Glance (Units):	Unit 1: Introduction to Computer Careers (2 weeks) Unit 2: Big Idea I: Creative Development Unit 3: Big Idea II: Data Unit 4: Big Idea III: Algorithms and Programming Unit 5: Big Idea IV: Computer Systems and Networks Unit 6: Big Idea V: Impact of Computing

Units

Unit Number and Title:	Unit 1: Introduction to computer careers
Duration:	2 weeks
Resource(s):	Online Career Databases Technology Resources Software: word processing, spreadsheet, presentation
Overview	To provide students with educational and career path options during and after high school.
Learning Goals	
Standard(s):	State of Connecticut Curriculum Frameworks Connecticut State Standards are met in the following areas: EKS.03.01 - Demonstrate use of relational expressions such as: equal to, not equal, greater than, less than, etc. EKS.05.02 - Analyze elements of a problem to develop creative solutions. EKS.05.04 Create ideas, proposals, and solutions to problems 21st Century Skills/International Society for Technology in Education National Business Education Association (NBEA) Standards
Essential Question(s):	● What challenges do computer professionals face in today's world? ● What will the computer industry look like in 10 years?
Enduring Understanding(s):	● Define different careers associated with computers. ● Define the requirements and education necessary for these careers.
Learning Goal(s): *Students will able to use their*	Students will be able to: ● Determine a career path after high school:

learning to:	○ Colleges / Universities / Traid schools / apprenticeships that offer careers. ○ Degree(s)/ Certifications required for career ○ Payscale ○ Work environment

Unit Number and Title:	Unit 2: Big Idea 1: Creative Development (CRD)
Duration:	Throughout the course
Resource(s):	External Resources: Collaboration Tools from Cornell University Center for Teaching Innovation 4 Methods to Enhance Student Collaboration in the Classroom from Concordia UniversityPortland
Unit Overview:	Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.
Learning Goals	
Standard(s):	CRD-1 CRD-1.A CRD-1.A.1 Incorporating multiple perspectives through collaboration improves computing innovations as they are developed CRD-2 Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/ feedback cycles, and that leaves ample room for experimentation and risk-taking.
Essential Question(s):	● How has working collaboratively with other students improved an overall project? ● What are some ways you can collect additional feedback on your program to use for improvements? ● What are some ways you currently plan your work before starting a project? ● What apps or programs have you stopped using because you didn't like the design of how you interacted with it?
Enduring Understanding(s):	Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.
Learning Goal(s): *Students will be able to use their learning to:*	Students will be able to: ● explain how collaboration affects the development of a solution. ● collaborate in the development of solutions (not assessed).

	● program function and purpose. ● investigate the situation, context, or task. ● generalize data sources through variables. ● explain how a code segment or program functions. ● determine and design an appropriate method or approach to achieve the purpose. ● explain how collaboration affects the development of a solution. ● explain how a code segment or program functions. ● acknowledge the intellectual property of others (not assessed). ● identifying and correcting errors. ● determine and design an appropriate method or approach to achieve the purpose. ● identify and correct errors in algorithms and programs, including error discovery through testing.

Unit Number and Title:	Unit 3: Big Idea 2: Data
Duration:	Throughout
Resource(s):	External Resources: Binary Numbers from CS Unplugged Blown to Bits- Chapter 1
Unit Overview:	Programs can be used to process data, which allows users to discover information and create new knowledge.
Learning Goals	
Standard(s):	DAT 1 The way a computer represents data internally is different from the way the data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people. DAT 2 Programs can be used to process data, which allows users to discover information and create new knowledge.
Essential Question(s):	● How can we use 1s and 0s to represent something complex like a video of the marching band playing a song? ● How can you predict the attendance at a school event using data gathered from social media? ● When is it more appropriate to use a computer to analyze data than to complete the analysis by hand?
Enduring Understanding(s):	The way a computer represents data internally is different from the way the data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.
Learning Goal(s): *Students will be able to use their learning to:*	Students will be able to: ● evaluate solution options. ● implement and apply an algorithm. ● explain how abstraction manages complexity. ● extract information from data.

	● explain how knowledge can be generated from data. ● describe the impact of gathering data. ● implement and apply an algorithm. ● explain how knowledge can be generated from data.

Unit Number and Title:	Unit 4: Big Idea 3 Algorithms and Programming
Duration:	Throughout the course
Resource(s):	External Resources: Collaboration Tools from Cornell University Center for Teaching Innovation 4 Methods to Enhance Student Collaboration in the Classroom from Concordia UniversityPortland
Unit Overview:	To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.
Learning Goals	
Standard(s):	AAP-1 To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways. AAP-1 To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways. AAP-2 The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-2 The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. AAP-3 Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

	AAP-4 There exist problems that computers cannot solve, and even when a computer can solve a problem, it may not be able to do so in a reasonable amount of time.
Essential Question(s):	● How can we store data in a program to solve problems? ● What might happen if you completed the steps in your regular morning routine to get ready and go to school in a different order? ● How might the reordering affect the decisions you make each morning? ● How do video games group the different actions for a player based on what key is pressed on the keyboard or controller? ● How do apps group different actions together based on user interaction, such as pressing buttons? ● What types of problems can be solved more easily with a computer, and what types can be solved more easily without a computer? Why?
Enduring Understanding(s):	● To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways. ● To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways ● The way statements are sequenced and combined in a program determines the computed result. ● Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence. ● There exist problems that computers cannot solve, and even when a computer can solve a problem, it may not be able to do so in a reasonable amount of time.
Learning Goal(s): *Students will be able to use their learning to:*	Students will be able to: ●

Unit Number and Title:	Unit 5: Big Idea 4 Computer Systems and Networks
Duration:	Throughout the course
Resource(s):	External Resources: Collaboration Tools from Cornell University Center for Teaching Innovation 4 Methods to Enhance Student Collaboration in the Classroom from Concordia UniversityPortland
Unit Overview:	While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.
Learning Goals	
Standard(s):	CSN-1 Computer systems and networks facilitate the transfer of data. CSN-2 Parallel and distributed computing leverage multiple computers to more quickly solve complex problems or process large data sets.
Essential Question(s):	● Why are long text messages sometimes delivered out of order? ● When an Internet service outage occurs in a different part of your town or city, how are you still able to access the Internet? ● What are the benefits of dividing tasks among group members? ● Is there a point where adding another group member would not make completing the task faster? Why
Enduring Understanding(s):	● Computer systems and networks facilitate the transfer of data. ● Parallel and distributed computing leverage multiple computers to more quickly solve complex problems or process large data sets.
Learning Goal(s): *Students will be able to use their learning to:*	Students will be able to: ● explain how computing systems work. ● evaluate solution options. ● explain how computing systems work.

	● parallel and distributed computing. ● evaluate solution options.

Unit Number and Title:	Unit 6: Big Idea 5 - Impact of Computing
Duration:	2 weeks
Resource(s):	External Resources: Collaboration Tools from Cornell University Center for Teaching Innovation 4 Methods to Enhance Student Collaboration in the Classroom from Concordia UniversityPortland
Unit Overview:	Exploring current innovation(s) in computer science and how does that benefit society
Learning Goals	
Standard(s):	IOC-1 While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences. IOC-2 The use of computing innovations may involve risks to personal safety and identity
Essential Question(s):	● What app or computer software do you use most often and would have a hard time going without? How does this software solve a problem for you or benefit you? ● Are innovators responsible for the harmful effects of their computing innovations, even if those effects were unintentional? Why or why not? ● What data is generated by smartphones, and what are they being used for?
Enduring Understanding(s):	● While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences. ● The use of computing innovations may involve risks to personal safety and identity
Learning Goal(s): *Students will be able to use their learning to:*	Students will be able to: ● describe the impact of a computing innovation. ● explain the digital divide. ● describe the impact of a computing innovation. ● explain computing bias. ● evaluate the use of computing based on legal and ethical factors.

	● explain how collaboration affects the development of a solution. ● explain legal and ethical concerns. ● evaluate the use of computing based on legal and ethical factors. ● describe the impact of gathering data. ● evaluate the use of computing based on legal and ethical factors.