

**FOREST LAKE AREA SCHOOLS  
FOREST LAKE, MN 55025**

**April 4, 2019**

**AGENDA ITEM: 9.12**

**TOPIC: Recommendation to Approve New – AP Computer Science Principles A/B**

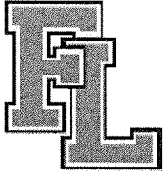
**BACKGROUND:**

In an effort to expand the opportunities for students to earn college credit through coursework at Forest Lake Area High School, teachers and administrators explored potential departments and course offerings for consideration.

**PROCESS:**

The data gathered through our work with Equal Opportunity Schools revealed that there is much interest in computer science among our high school students. It was determined that *AP Computer Science Principles* would be a good fit, as the content is appropriate for 9-12<sup>th</sup> grade students, and it provides an additional college credit bearing option for students through the Math department.

**RECOMMENDATION:** Approval of New Course *AP Computer Science Principles A/B*  
*\*To be offered in 19-20 pending Board Approval*



## Request for Curriculum Change New Course Proposal

Date: 02/18/19

Department/Subject: Math/Computer Science

Person(s) initiating the request: Brad Ward

Department members involved in the development of the course proposal: Brian Roy

School(s): Forest Lake Area High School

Grade(s): 9-12

Course Title: **Advanced Placement Computer Science Principles A,B**

Describe the process that led to this request for a new course:

In 2016, the College Board approved AP CS Principles as an additional AP opportunity in the field of computer science. This course was developed with the knowledge that the current AP Computer Science course lacks enrollment diversity in a career area that is growing quickly. In less than two years, AP CS Principles has overtaken AP CS in enrollment and exams administered.

Unlike APCS, APCS Principles is open to students in grades 9-12 thus opening another avenue for students in 9th and 10th grade to enroll in an AP course. The course is constructed to serve as a gateway for future programming class opportunities. Code.org has a College Board approved curriculum that is relevant and with the appropriate rigor to provide students and teachers a rich experience.

## Specific Information to Accompany Request

### Course Outline/Learner Outcomes

Complete a [Course Outline](#) for your proposed course and submit it with your proposal.

### Budget Considerations

Check any of the following that will be needed for your proposed course and indicate the anticipated cost for each:

\_\_\_ **Textbooks**

Title(s): [Blown to Bits](#) (optional)

Approximate total cost: Available for free pdf download or purchased as bound text.

\_\_\_ **Digital Curriculum Resources**

Title(s):

Approximate total cost: All course materials are free provided instructor takes 1 week training form Code.org

One-time cost or annual renewal? No plans to add payroll.

\_\_\_ **Other Curriculum Materials (consumables, supplies, ancillaries, etc.)**

Materials: Miscellaneous supplies

Consumable/non-consumable?

Approximate total cost:

\_\_\_ **Technology Devices/Equipment/Hardware**

Devices/equipment needed: Students need a computer and web access.

Approximate cost:

\_\_\_ **Staff Development**

Staff Development description: Attend free one week summer training, plus participate in PLC cohort with other code.org APCS Principles teachers.

Approximate cost:

Frequency (one time? yearly?):

## Effects of the New Course Proposal

Provide a brief explanation for any of the following the new course proposal will affect:

\_\_\_ Current curriculum:

- Current proposal would replace C++ with APCS Principles. Teacher indicates that the courses have similar goals.

\_\_\_ Programs in other departments/grade levels:

- Course would be offered via math department which also offers APCS A/B and C++. The course is also able to be taught by CTE certified teacher provided they receive proper training.

\_\_\_ Credit requirements:

- 1.0 credits (2 semester course)

\_\_\_ Prerequisite requirements:

- None. Open to students 9-12 grade who are interested in programming.

\_\_\_ Staffing:

- Requires math or CTE license.

\_\_\_ Space allocation:

\_\_\_ Master schedule:

\_\_\_ District Technology impact:

\_\_\_ Other:

## Signatures

*Signatures indicate agreement with the course proposal.*

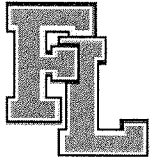
Jim Caldwell  
Administrative Department Lead

John-Paul Jacobson  
Forest Lake Area Middle School Principal

Kelly Lafean  
Area Learning Center Principal

Jim Caldwell  
High School Principal

Diane Giorgi  
Director of Teaching and Learning



# Secondary Course Outline

## Forest Lake Area Schools

**Course name: Advanced Placement Computer Science Principles A,B**

**Grade(s): 9-12**

**Course description (as found in the registration guide):**

The course covers a broad range of foundational topics such as programming, algorithms, the Internet, big data, digital privacy and security, and the societal impacts of computing. The course contains five core units of study, with a sixth unit devoted almost exclusively to students working on their *AP Performance Task*(PT) projects. Each unit has one or two “chapters” of related lessons that usually conclude with some kind of project or summative assessment. This course is intended to be suitable as a first course in computing though students with a variety of backgrounds and prior experiences will also find the course engaging and with plenty of challenges.

**Graduation standard(s) information: (alignment with Minnesota Academic Standards or national standards)**

**Coverage of the AP CS Principles Framework and Computational Thinking Practices**

The CS Principles Framework outlines seven “Big Ideas” of computing, and six “Computational Thinking Practices”. Activities in the course should ensure that students are engaging in the Computational Thinking Practices while investigating the Big Ideas.

rev. April 2016

**Seven Big Ideas**

*The [course is] organized around seven big ideas, which encompass ideas foundational to studying computer science.*

Big Idea 1: Creativity

Big Idea 2: Abstraction

Big Idea 3: Data

Big Idea 4: Algorithms

Big Idea 5: Programming

Big Idea 6: The Internet

Big Idea 7: Global Impacts

**Six Computational Thinking Practices**

*Computational thinking practices capture important aspects of the work that computer scientists engage in.*

P1: Connecting Computing

P2: Creating Computational Artifacts

P3: Abstracting

P4: Analyzing Problems and Artifacts

P5: Communicating

P6: Collaborating

**Learner outcomes:**

# Computational Thinking Practices

## P1: Connecting Computing

Developments in computing have far-reaching effects on society and have led to significant innovations. The developments have implications for individuals, society, commercial markets, and innovation. Students in this course study these effects, and they learn to draw connections between different computing concepts. Students are expected to:

- ▶ Identify impacts of computing.
- ▶ Describe connections between people and computing.
- ▶ Explain connections between computing concepts.

## P2: Creating Computational Artifacts

Computing is a creative discipline in which creation takes many forms, such as remixing digital music, generating animations, developing websites, and writing programs. Students in this course engage in the creative aspects of computing by designing and developing interesting computational artifacts as well as by applying computing techniques to creatively solve problems. Students are expected to:

- ▶ Create a computational artifact with a practical, personal, or societal intent.
- ▶ Select appropriate techniques to develop a computational artifact.
- ▶ Use appropriate algorithmic and information management principles.

## P3: Abstracting

Computational thinking requires understanding and applying abstraction at multiple levels, such as privacy in social networking applications, logic gates and bits, and the human genome project. Students in this course use abstraction to develop models and simulations of natural and artificial phenomena, use them to make predictions about the world, and analyze their efficacy and validity. Students are expected to:

- ▶ Explain how data, information, or knowledge is represented for computational use.
- ▶ Explain how abstractions are used in computation or modeling.
- ▶ Identify abstractions.
- ▶ Describe modeling in a computational context.

## P4: Analyzing Problems and Artifacts

The results and artifacts of computation and the computational techniques and strategies that generate them can be understood intrinsically both for what they are as well as for what they produce. They can also be analyzed and evaluated by applying aesthetic, mathematical, pragmatic, and other criteria. Students in this course design and produce solutions, models, and artifacts, and they evaluate and analyze their own computational work as well as the computational work others have produced. Students are expected to:

- ▶ Evaluate a proposed solution to a problem.
- ▶ Locate and correct errors.
- ▶ Explain how an artifact functions.
- ▶ Justify appropriateness and correctness of a solution, model, or artifact.

## P5: Communicating

Students in this course describe computation and the impact of technology and computation, explain and justify the design and appropriateness of their computational choices, and analyze and describe both computational artifacts and the results or behaviors of such artifacts. Communication includes written and oral descriptions supported by graphs, visualizations, and computational analysis. Students are expected to:

- ▶ Explain the meaning of a result in context.
- ▶ Describe computation with accurate and precise language, notations, or visualizations.
- ▶ Summarize the purpose of a computational artifact.

## P6: Collaborating

Innovation can occur when people work together or independently. People working collaboratively can often achieve more than individuals working alone. Learning to collaborate effectively includes drawing on diverse perspectives, skills, and the backgrounds of peers to address complex and open-ended problems. Students in this course collaborate on a number of activities, including the investigation of questions using data sets and the production of computational artifacts. Students are expected to:

- ▶ Collaborate with another student in solving a computational problem.
- ▶ Collaborate with another student in producing an artifact.
- ▶ Share the workload by providing individual contributions to an overall collaborative effort.
- ▶ Foster a constructive, collaborative climate by resolving conflicts and facilitating the contributions of a partner or team member.
- ▶ Exchange knowledge and feedback with a partner or team member.
- ▶ Review and revise their work as needed to create a high-quality artifact.

**Course content: (Write this in outline format.)**

## Big Idea 1: Creativity

**Essential Questions:**

- ▶ How can a creative development process affect the creation of computational artifacts?
- ▶ How can computing and the use of computational tools foster creative expression?
- ▶ How can computing extend traditional forms of human expression and experience?

Enduring Understandings(Students will understand that ... )	Learning Objectives(Students will be able to ... )	Essential Knowledge (Students will know that ... )
<b>EU 1.1</b> Creative development can be an essential process for creating computational artifacts.	<b>IO 1.1.1</b> Apply a creative development process when creating computational artifacts. [P2]	<b>EK 1.1.1A</b> A creative process in the development of a computational artifact can include, but is not limited to, employing nontraditional, nonprescribed techniques; the use of novel combinations of artifacts, tools, and techniques; and the exploration of personal curiosities. <b>EK 1.1.1B</b> Creating computational artifacts employs an iterative and often exploratory process to translate ideas into tangible form.
<b>EU 1.2</b> Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.	<b>IO 1.2.1</b> Create a computational artifact for creative expression. [P2]	<b>EK 1.2.1A</b> A computational artifact is something created by a human using a computer and can be, but is not limited to, a program, an image, an audio, a video, a presentation, or a Web page file. <b>EK 1.2.1B</b> Creating computational artifacts requires understanding of and use of software tools and services. <b>EK 1.2.1C</b> Computing tools and techniques are used to create computational artifacts and can include, but are not limited to, programming integrated development environments (IDEs), spreadsheets, three-dimensional (3-D) printers, or text editors. <b>EK 1.2.1D</b> A creatively developed computational artifact can be created by using nontraditional, nonprescribed computing techniques. <b>EK 1.2.1E</b> Creative expressions in a computational artifact can reflect personal expressions of ideas or interests.
	<b>IO 1.2.2</b> Create a computational artifact using computing tools and techniques to solve a problem. [P2]	<b>EK 1.2.2A</b> Computing tools and techniques can enhance the process of finding a solution to a problem. <b>EK 1.2.2B</b> A creative development process for creating computational artifacts can be used to solve problems when

		traditional or prescribed computing techniques are not effective.
	<b>IO 1.2.3</b> Create a new computational artifact by combining or modifying existing artifacts. [P2]	<b>Ek 1.2.3A</b> Creating computational artifacts can be done by combining and modifying existing artifacts or by creating new artifacts. <b>Ek 1.2.3B</b> Computation facilitates the creation and modification of computational artifacts with enhanced detail and precision. <b>Ek 1.2.3C</b> Combining or modifying existing artifacts can show personal expression of ideas.
	<b>IO 1.2.4</b> Collaborate in the creation of computational artifacts. [P6]	<b>Ek 1.2.4A</b> A collaboratively created computational artifact reflects effort by more than one person. <b>Ek 1.2.4B</b> Effective collaborative teams consider the use of online collaborative tools. <b>Ek 1.2.4C</b> Effective collaborative teams practice interpersonal communication, consensus building, conflict resolution, and negotiation. <b>Ek 1.2.4D</b> Effective collaboration strategies enhance performance. <b>Ek 1.2.4E</b> Collaboration facilitates the application of multiple perspectives (including sociocultural perspectives) and diverse talents and skills in developing computational artifacts. <b>Ek 1.2.4F</b> A collaboratively created computational artifact can reflect personal expressions of ideas.
	<b>IO 1.2.5</b> Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4]	<b>Ek 1.2.5A</b> The context in which an artifact is used determines the correctness, usability, functionality, and suitability of the artifact. <b>Ek 1.2.5B</b> A computational artifact may have weaknesses, mistakes, or errors depending on the type of artifact. <b>Ek 1.2.5C</b> The functionality of a computational artifact may be related to how it is used or perceived. <b>Ek 1.2.5D</b> The suitability (or appropriateness) of a computational artifact may be related to how it is used or perceived.
<b>EU 1.3</b> Computing can extend traditional forms of human expression and experience.	<b>IO 1.3.1</b> Use computing tools and techniques for creative expression. [P2]	<b>Ek 1.3.1A</b> Creating digital effects, images, audio, video, and animations has transformed industries. <b>Ek 1.3.1B</b> Digital audio and music can be created by synthesizing sounds, sampling existing audio and music, and recording and manipulating sounds, including layering and looping. <b>Ek 1.3.1C</b> Digital images can be created by generating pixel patterns, manipulating existing digital images, or combining images. <b>Ek 1.3.1D</b> Digital effects and animations can be created by using existing software or modified software that includes functionality to implement the effects and animations. <b>Ek 1.3.1E</b> Computing enables creative exploration of both real and virtual phenomena.

## Big Idea 2: Abstraction

### Essential Questions:

- ▶ How are vastly different kinds of data, physical phenomena, and mathematical concepts represented on a computer?
- ▶ How does abstraction help us in writing programs, creating computational artifacts, and solving problems?
- ▶ How can computational models and simulations help generate new understanding and knowledge?

Enduring Understandings(Students will understand that ... )	Learning Objectives(Students will be able to ... )	Essential Knowledge (Students will know that ... )
---	--	--



<p><b>EU 2.1</b> A variety of abstractions built on binary sequences can be used to represent all digital data.</p>	<p><b>IO 2.1.1</b> Describe the variety of abstractions used to represent data. [P3]</p>	<p><b>Ek 2.1.1A</b> Digital data is represented by abstractions at different levels.  <b>Ek 2.1.1B</b> At the lowest level, all digital data are represented by bits.  <b>Ek 2.1.1C</b> At a higher level, bits are grouped to represent abstractions, including but not limited to numbers, characters, and color.  <b>Ek 2.1.1D</b> Number bases, including binary, decimal, and hexadecimal, are used to represent and investigate digital data.  <b>Ek 2.1.1E</b> 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.  <b>EXCLUSION STATEMENT (for Ek 2.1.1E):</b> Two's complement conversions are beyond the scope of this course and the AP Exam.  <b>Ek 2.1.1F</b> Hexadecimal (base 16) is used to represent digital data because hexadecimal representation uses fewer digits than binary.  <b>Ek 2.1.1G</b> Numbers can be converted from any base to any other base.</p>
	<p><b>IO 2.1.2</b> Explain how binary sequences are used to represent digital data. [P5]</p>	<p><b>Ek 2.1.2A</b> A finite representation is used to model the infinite mathematical concept of a number.  <b>EXCLUSION STATEMENT (for Ek 2.1.2A):</b> Binary representations of scientific notation are beyond the scope of this course and the AP Exam.  <b>Ek 2.1.2B</b> In many programming languages, the fixed number of bits used to represent characters or integers limits the range of integer values and mathematical operations; this limitation can result in overflow or other errors.  <b>EXCLUSION STATEMENT (for Ek 2.1.2B):</b> Range limitations of any one language, compiler, or architecture are beyond the scope of this course and the AP Exam.  <b>Ek 2.1.2C</b> In many programming languages, the fixed number of bits used to represent real numbers (as floating-point numbers) limits the range of floating-point values and mathematical operations; this limitation can result in round-off and other errors.  <b>Ek 2.1.2D</b> The interpretation of a binary sequence depends on how it is used.  <b>Ek 2.1.2E</b> A sequence of bits may represent instructions or data.  <b>Ek 2.1.2F</b> A sequence of bits may represent different types of data in different contexts.</p>
<p><b>EU 2.2</b> Multiple levels of abstraction are used to write programs or create other computational artifacts.</p>	<p><b>IO 2.2.1</b> Develop an abstraction when writing a program or creating other computational artifacts. [P2]</p>	<p><b>Ek 2.2.1A</b> The process of developing an abstraction involves removing detail and generalizing functionality.  <b>Ek 2.2.1B</b> An abstraction extracts common features from specific examples in order to generalize concepts.  <b>Ek 2.2.1C</b> An abstraction generalizes functionality with input parameters that allow software reuse.  <b>EXCLUSION STATEMENT (for Ek 2.2.1C):</b> An understanding of the difference between value and reference parameters is beyond the scope of this course and the AP Exam.</p>
	<p><b>IO 2.2.2</b> Use multiple levels of abstraction to write programs. [P3]</p>	<p><b>Ek 2.2.2A</b> Software is developed using multiple levels of abstractions, such as constants, expressions, statements, procedures, and libraries.  <b>Ek 2.2.2B</b> Being aware of and using multiple levels of abstractions in developing programs help to more effectively apply available resources and tools to solve problems.</p>
	<p><b>IO 2.2.3</b> Identify multiple levels of abstractions that are used when writing programs. [P3]</p>	<p><b>Ek 2.2.3A</b> Different programming languages offer different levels of abstraction.  <b>EXCLUSION STATEMENT (for Ek 2.2.3A):</b> Knowledge of the abstraction capabilities of all programming languages is beyond the scope of this course and the AP Exam.  <b>Ek 2.2.3B</b> High-level programming languages provide more</p>

		<p>abstractions for the programmer and make it easier for people to read and write a program.</p> <p><b>Ek 2.2.3C</b> Code in a programming language is often translated into code in another (lower-level) language to be executed on a computer.</p> <p><b>Ek 2.2.3D</b> In an abstraction hierarchy, higher levels of abstraction (the most general concepts) would be placed toward the top and lower-level abstractions (the more specific concepts) toward the bottom.</p> <p><b>Ek 2.2.3E</b> Binary data is processed by physical layers of computing hardware, including gates, chips, and components.</p> <p><b>Ek 2.2.3F</b> A logic gate is a hardware abstraction that is modeled by a Boolean function.</p> <p><b>EXCLUSION STATEMENT (for Ek 2.2.3F):</b> Memorization of specific gate visual representations is beyond the scope of this course and the AP Exam.</p> <p><b>Ek 2.2.3G</b> A chip is an abstraction composed of low-level components and circuits that perform a specific function.</p> <p><b>Ek 2.2.3H</b> A hardware component can be low level like a transistor or high level like a video card.</p> <p><b>Ek 2.2.3I</b> Hardware is built using multiple levels of abstractions, such as transistors, logic gates, chips, memory, motherboards, special purpose cards, and storage devices.</p> <p><b>Ek 2.2.3J</b> Applications and systems are designed, developed, and analyzed using levels of hardware, software, and conceptual abstractions.</p> <p><b>Ek 2.2.3k</b> Lower-level abstractions can be combined to make higher-level abstractions, such as short message services (SMS) or email messages, images, audio files, and videos.</p>
<p><b>EU 2.3</b> Models and simulations use abstraction to generate new understanding and knowledge.</p>	<p><b>IO 2.3.1</b> Use models and simulations to represent phenomena. [P3]</p>	<p><b>Ek 2.3.1A</b> Models and simulations are simplified representations of more complex objects or phenomena.</p> <p><b>Ek 2.3.1B</b> Models may use different abstractions or levels of abstraction depending on the objects or phenomena being posed.</p> <p><b>Ek 2.3.1C</b> Models often omit unnecessary features of the objects or phenomena that are being modeled.</p> <p><b>Ek 2.3.1D</b> Simulations mimic real-world events without the cost or danger of building and testing the phenomena in the real world.</p>
	<p><b>IO 2.3.2</b> Use models and simulations to formulate, refine, and test hypotheses. [P3]</p>	<p><b>Ek 2.3.2A</b> Models and simulations facilitate the formulation and refinement of hypotheses related to the objects or phenomena under consideration.</p> <p><b>Ek 2.3.2B</b> Hypotheses are formulated to explain the objects or phenomena being modeled.</p> <p><b>Ek 2.3.2C</b> Hypotheses are refined by examining the insights that models and simulations provide into the objects or phenomena.</p> <p><b>Ek 2.3.2D</b> The results of simulations may generate new knowledge and new hypotheses related to the phenomena being modeled.</p> <p><b>Ek 2.3.2E</b> Simulations allow hypotheses to be tested without the constraints of the real world.</p> <p><b>Ek 2.3.2F</b> Simulations can facilitate extensive and rapid testing of models.</p> <p><b>Ek 2.3.2G</b> The time required for simulations is impacted by the level of detail and quality of the models and the software and hardware used for the simulation.</p> <p><b>Ek 2.3.2H</b> Rapid and extensive testing allows models to be changed to accurately reflect the objects or phenomena being modeled.</p>

## Big Idea 3: Data and Information

Essential Questions:

- ▶ How can computation be employed to help people process data and information to gain insight and knowledge?
- ▶ How can computation be employed to facilitate exploration and discovery when working with data?
- ▶ What considerations and trade-offs arise in the computational manipulation of data?
- ▶ What opportunities do large data sets provide for solving problems and creating knowledge?

Enduring Understandings(Students will understand that ... )	Learning Objectives(Students will be able to ... )	Essential Knowledge (Students will know that ... )
<p><b>EU 3.1</b> People use computer programs to process information to gain insight and knowledge.</p>	<p><b>IO 3.1.1</b> Find patterns and test hypotheses about digitally processed information to gain insight and knowledge. [P4]</p>	<p><b>Ek 3.1.1A</b> Computers are used in an iterative and interactive way when processing digital information to gain insight and knowledge.  <b>Ek 3.1.1B</b> Digital information can be filtered and cleaned by using computers to process information.  <b>Ek 3.1.1C</b> Combining data sources, clustering data, and data classification are part of the process of using computers to process information.  <b>Ek 3.1.1D</b> Insight and knowledge can be obtained from translating and transforming digitally represented information.  <b>Ek 3.1.1E</b> Patterns can emerge when data is transformed using computational tools.</p>
	<p><b>IO 3.1.2</b> Collaborate when processing information to gain insight and knowledge. [P6]</p>	<p><b>Ek 3.1.2A</b> Collaboration is an important part of solving data- driven problems.  <b>Ek 3.1.2B</b> Collaboration facilitates solving computational problems by applying multiple perspectives, experiences, and skill sets.  <b>Ek 3.1.2C</b> Communication between participants working on data-driven problems gives rise to enhanced insights and knowledge.  <b>Ek 3.1.2D</b> Collaboration in developing hypotheses and questions, and in testing hypotheses and answering questions, about data helps participants gain insight and knowledge.  <b>Ek 3.1.2E</b> Collaborating face-to-face and using online collaborative tools can facilitate processing information to gain insight and knowledge.  <b>Ek 3.1.2F</b> Investigating large data sets collaboratively can lead to insight and knowledge not obtained when working alone.</p>
	<p><b>IO 3.1.3</b> Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language. [P5]</p>	<p><b>Ek 3.1.3A</b> Visualization tools and software can communicate information about data.  <b>Ek 3.1.3B</b> Tables, diagrams, and textual displays can be used in communicating insight and knowledge gained from data.  <b>Ek 3.1.3C</b> Summaries of data analyzed computationally can be effective in communicating insight and knowledge gained from digitally represented information.  <b>Ek 3.1.3D</b> Transforming information can be effective in communicating knowledge gained from data.  <b>Ek 3.1.3E</b> Interactivity with data is an aspect of communicating.</p>
<p><b>EU 3.2</b> Computing facilitates exploration and the discovery of connections in information.</p>	<p><b>IO 3.2.1</b> Extract information from data to discover and explain connections or trends. [P1]</p>	<p><b>Ek 3.2.1A</b> Large data sets provide opportunities and challenges for extracting information and knowledge.  <b>Ek 3.2.1B</b> Large data sets provide opportunities for identifying trends, making connections in data, and solving problems.  <b>Ek 3.2.1C</b> Computing tools facilitate the discovery of connections in information within large data sets.  <b>Ek 3.2.1D</b> Search tools are essential for efficiently finding information.  <b>Ek 3.2.1E</b> Information filtering systems are important tools for finding information and recognizing patterns in the information.  <b>Ek 3.2.1F</b> Software tools, including spreadsheets and databases, help to efficiently organize and find trends</p>

		<p>in information.</p> <p><b>EXCLUSION STATEMENT (for Ek 3.2.1F):</b> Students are not expected to know specific formulas or options available in spreadsheet or database software packages.</p> <p><b>Ek 3.2.1G</b> Metadata is data about data.</p> <p><b>Ek 3.2.1H</b> Metadata can be descriptive data about an image, a Web page, or other complex objects.</p> <p><b>Ek 3.2.1I</b> Metadata can increase the effective use of data or data sets by providing additional information about various aspects of that data.</p>
	<p><b>IO 3.2.2</b> Determine how large data sets impact the use of computational processes to discover information and knowledge. [P3]</p>	<p><b>Ek 3.2.2A</b> Large data sets include data such as transactions, measurements, texts, sounds, images, and videos.</p> <p><b>Ek 3.2.2B</b> The storing, processing, and curating of large data sets is challenging.</p> <p><b>Ek 3.2.2C</b> Structuring large data sets for analysis can be challenging.</p> <p><b>Ek 3.2.2D</b> Maintaining privacy of large data sets containing personal information can be challenging.</p> <p><b>Ek 3.2.2E</b> Scalability of systems is an important consideration when data sets are large.</p> <p><b>Ek 3.2.2F</b> The size or scale of a system that stores data affects how that data set is used.</p> <p><b>Ek 3.2.2G</b> The effective use of large data sets requires computational solutions.</p> <p><b>Ek 3.2.2H</b> Analytical techniques to store, manage, transmit, and process data sets change as the size of data sets scale.</p>

## Big Idea 4: Algorithms

### Essential Questions:

- ▶ How are algorithms implemented and executed on computers and computational devices?
- ▶ Why are some languages better than others when used to implement algorithms?
- ▶ What kinds of problems are easy, what kinds are difficult, and what kinds are impossible to solve algorithmically?
- ▶ How are algorithms evaluated?

Enduring Understandings(Students will understand that ... )	Learning Objectives(Students will be able to ... )	Essential Knowledge (Students will know that ... )
<p><b>EU 4.1</b> Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages.</p>	<p><b>IO 4.1.1</b> Develop an algorithm for implementation in a program. [P2]</p>	<p><b>Ek 4.1.1A</b> Sequencing, selection, and iteration are building blocks of algorithms.</p> <p><b>Ek 4.1.1B</b> Sequencing is the application of each step of an algorithm in the order in which the statements are given.</p> <p><b>Ek 4.1.1C</b> Selection uses a Boolean condition to determine which of two parts of an algorithm is used.</p> <p><b>Ek 4.1.1D</b> Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times.</p> <p><b>Ek 4.1.1E</b> Algorithms can be combined to make new algorithms.</p> <p><b>Ek 4.1.1F</b> Using existing correct algorithms as building blocks for constructing a new algorithm helps ensure the new algorithm is correct.</p> <p><b>Ek 4.1.1G</b> Knowledge of standard algorithms can help in constructing new algorithms.</p> <p><b>Ek 4.1.1H</b> Different algorithms can be developed to solve the same problem.</p> <p><b>Ek 4.1.1I</b> Developing a new algorithm to solve a problem can yield insight into the problem.</p>
	<p><b>IO 4.1.2</b> Express an algorithm in a language. [P5]</p>	<p><b>Ek 4.1.2A</b> Languages for algorithms include natural language, pseudocode, and visual and textual</p>

		<p>programming languages.</p> <p><b>Ek 4.1.2B</b> Natural language and pseudocode describe algorithms so that humans can understand them.</p> <p><b>Ek 4.1.2C</b> Algorithms described in programming languages can be executed on a computer.</p> <p><b>Ek 4.1.2D</b> Different languages are better suited for expressing different algorithms.</p> <p><b>Ek 4.1.2E</b> Some programming languages are designed for specific domains and are better for expressing algorithms in those domains.</p> <p><b>Ek 4.1.2F</b> The language used to express an algorithm can affect characteristics such as clarity or readability but not whether an algorithmic solution exists.</p> <p><b>Ek 4.1.2G</b> Every algorithm can be constructed using only sequencing, selection, and iteration.</p> <p><b>Ek 4.1.2H</b> Nearly all programming languages are equivalent in terms of being able to express any algorithm.</p> <p><b>Ek 4.1.2I</b> Clarity and readability are important considerations when expressing an algorithm in a language.</p>
<p><b>EU 4.2</b> Algorithms can solve many, but not all, computational problems.</p>	<p><b>IO 4.2.1</b> Explain the difference between algorithms that run in a reasonable time and those that do not run in a reasonable time. [P1]</p> <p><b>EXCIUSION STATEmENT (for IO 4.2.1):</b> Any discussion of nondeterministic polynomial (NP) is beyond the scope of this course and the AP Exam.</p>	<p><b>Ek 4.2.1A</b> Many problems can be solved in a reasonable time. <b>Ek 4.2.1B</b> Reasonable time means that the number of steps the algorithm takes is less than or equal to a polynomial function (constant, linear, square, cube, etc.) of the size of the input.</p> <p><b>EXCIUSION STATEmENT (for Ek 4.2.1B):</b> Using nonpolynomial functions to describe relationships between the number of steps required by an algorithm and the input size is beyond the scope of this course and the AP Exam.</p> <p><b>Ek 4.2.1C</b> Some problems cannot be solved in a reasonable time, even for small input sizes.</p> <p><b>Ek 4.2.1D</b> Some problems can be solved but not in a reasonable time. In these cases, heuristic approaches may be helpful to find solutions in reasonable time.</p>
	<p><b>IO 4.2.2</b> Explain the difference between solvable and unsolvable problems in computer science. [P1]</p> <p><b>EXCIUSION STATEmENT (for IO 4.2.2):</b> Determining whether a given problem is solvable or unsolvable is beyond the scope of this course and the AP Exam.</p>	<p><b>Ek 4.2.2A</b> A heuristic is a technique that may allow us to find an approximate solution when typical methods fail to find an exact solution.</p> <p><b>Ek 4.2.2B</b> Heuristics may be helpful for finding an approximate solution more quickly when exact methods are too slow.</p> <p><b>EXCIUSION STATEmENT (for Ek 4.2.2B):</b> Specific heuristic solutions are beyond the scope of this course and the AP Exam.</p> <p><b>Ek 4.2.2C</b> Some optimization problems such as "find the best" or "find the smallest" cannot be solved in a reasonable time but approximations to the optimal solution can.</p> <p><b>Ek 4.2.2D</b> Some problems cannot be solved using any algorithm.</p>
	<p><b>IO 4.2.3</b> Explain the existence of undecidable problems in computer science. [P1]</p>	<p><b>Ek 4.2.3A</b> An undecidable problem may have instances that have an algorithmic solution, but there is no algorithmic solution that solves all instances of the problem.</p> <p><b>Ek 4.2.3B</b> A decidable problem is one in which an algorithm can be constructed to answer "yes" or "no" for all inputs (e.g., "Is the number even?").</p> <p><b>Ek 4.2.3C</b> An undecidable problem is one in which no algorithm can be constructed that always leads to a correct yes-or-no answer.</p> <p><b>EXCIUSION STATEmENT (for Ek 4.2.3C):</b> Determining whether a given problem is undecidable is beyond the scope of this course and the AP Exam.</p>

# Big Idea 5: Programming

## Essential Questions:

- ▶ How are programs developed to help people, organizations, or society solve problems?
- ▶ How are programs used for creative expression, to satisfy personal curiosity, or to create new knowledge?
- ▶ How do computer programs implement algorithms?
- ▶ How does abstraction make the development of computer programs possible?
- ▶ How do people develop and test computer programs?
- ▶ Which mathematical and logical concepts are fundamental to computer programming?

Enduring Understandings(Students will understand that ... )	Learning Objectives(Students will be able to ... )	Essential Knowledge (Students will know that ... )
<p><b>EU 5.1</b> Programs can be developed for creative expression, to satisfy personal curiosity, to create new knowledge, or to solve problems (to help people, organizations, or society).</p>	<p><b>IO 5.1.1</b> Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2]</p>	<p><b>Ek 5.1.1A</b> Programs are developed and used in a variety of ways by a wide range of people depending on the goals of the programmer.</p> <p><b>Ek 5.1.1B</b> Programs developed for creative expression, to satisfy personal curiosity, or to create new knowledge may have visual, audible, or tactile inputs and outputs.</p> <p><b>Ek 5.1.1C</b> Programs developed for creative expression, to satisfy personal curiosity, or to create new knowledge may be developed with different standards or methods than programs developed for widespread distribution.</p> <p><b>Ek 5.1.1D</b> Additional desired outcomes may be realized independently of the original purpose of the program.</p> <p><b>Ek 5.1.1E</b> A computer program or the results of running a program may be rapidly shared with a large number of users and can have widespread impact on individuals, organizations, and society.</p> <p><b>Ek 5.1.1F</b> Advances in computing have generated and increased creativity in other fields.</p>
	<p><b>IO 5.1.2</b> Develop a correct program to solve problems. [P2]</p>	<p><b>Ek 5.1.2A</b> An iterative process of program development helps in developing a correct program to solve problems.</p> <p><b>Ek 5.1.2B</b> Developing correct program components and then combining them helps in creating correct programs.</p> <p><b>Ek 5.1.2C</b> Incrementally adding tested program segments to correct working programs helps create large correct programs.</p> <p><b>Ek 5.1.2D</b> Program documentation helps programmers develop and maintain correct programs to efficiently solve problems.</p> <p><b>Ek 5.1.2E</b> Documentation about program components, such as code segments and procedures, helps in developing and maintaining programs.</p> <p><b>Ek 5.1.2F</b> Documentation helps in developing and maintaining programs when working individually or in collaborative programming environments.</p> <p><b>Ek 5.1.2G</b> Program development includes identifying programmer and user concerns that affect the solution to problems.</p> <p><b>Ek 5.1.2H</b> Consultation and communication with program users is an important aspect of program development to solve problems.</p> <p><b>Ek 5.1.2I</b> A programmer's knowledge and skill affects how a program is developed and how it is used to solve a problem.</p> <p><b>Ek 5.1.2J</b> A programmer designs, implements, tests, debugs, and maintains programs when solving problems.</p>

	<p><b>IO 5.1.3</b> Collaborate to develop a program. [P6]</p>	<p><b>Ek 5.1.3A</b> Collaboration can decrease the size and complexity of tasks required of individual programmers.</p> <p><b>Ek 5.1.3B</b> Collaboration facilitates multiple perspectives in developing ideas for solving problems by programming.</p> <p><b>Ek 5.1.3C</b> Collaboration in the iterative development of a program requires different skills than developing a program alone.</p> <p><b>Ek 5.1.3D</b> Collaboration can make it easier to find and correct errors when developing programs.</p> <p><b>Ek 5.1.3E</b> Collaboration facilitates developing program components independently.</p> <p><b>Ek 5.1.3F</b> Effective communication between participants is required for successful collaboration when developing programs.</p>
<p><b>EU 5.2</b> People write programs to execute algorithms.</p>	<p><b>IO 5.2.1</b> Explain how programs implement algorithms. [P3]</p>	<p><b>Ek 5.2.1A</b> Algorithms are implemented using program instructions that are processed during program execution.</p> <p><b>Ek 5.2.1B</b> Program instructions are executed sequentially.</p> <p><b>Ek 5.2.1C</b> Program instructions may involve variables that are initialized and updated, read, and written.</p> <p><b>Ek 5.2.1D</b> An understanding of instruction processing and program execution is useful for programming.</p> <p><b>Ek 5.2.1E</b> Program execution automates processes.</p> <p><b>Ek 5.2.1F</b> Processes use memory, a central processing unit (CPU), and input and output.</p> <p><b>Ek 5.2.1G</b> A process may execute by itself or with other processes.</p> <p><b>Ek 5.2.1H</b> A process may execute on one or several CPUs.</p> <p><b>Ek 5.2.1I</b> Executable programs increase the scale of problems that can be addressed.</p> <p><b>Ek 5.2.1J</b> Simple algorithms can solve a large set of problems when automated.</p> <p><b>Ek 5.2.1k</b> Improvements in algorithms, hardware, and software increase the kinds of problems and the size of problems solvable by programming.</p>
<p><b>EU 5.3</b> Programming is facilitated by appropriate abstractions.</p>	<p><b>IO 5.3.1</b> Use abstraction to manage complexity in programs. [P3]</p>	<p><b>Ek 5.3.1A</b> Procedures are reusable programming abstractions.</p> <p><b>Ek 5.3.1B</b> A procedure is a named grouping of programming instructions.</p> <p><b>Ek 5.3.1C</b> Procedures reduce the complexity of writing and maintaining programs.</p> <p><b>Ek 5.3.1D</b> Procedures have names and may have parameters and return values.</p> <p><b>Ek 5.3.1E</b> Parameterization can generalize a specific solution.</p> <p><b>Ek 5.3.1F</b> Parameters generalize a solution by allowing a procedure to be used instead of duplicated code.</p> <p><b>Ek 5.3.1G</b> Parameters provide different values as input to procedures when they are called in a program.</p> <p><b>Ek 5.3.1H</b> Data abstraction provides a means of separating behavior from implementation.</p> <p><b>Ek 5.3.1I</b> Strings and string operations, including concatenation and some form of substring, are common in many programs.</p> <p><b>Ek 5.3.1J</b> Integers and floating-point numbers are used in programs without requiring understanding of how they are implemented.</p> <p><b>Ek 5.3.1k</b> Lists and list operations, such as add, remove, and search, are common in many programs.</p> <p><b>Ek 5.3.1l</b> Using lists and procedures as abstractions in programming can result in programs that are easier to develop and maintain.</p> <p><b>Ek 5.3.1m</b> Application program interfaces (APIs) and libraries simplify complex programming tasks.</p>

		<p><b>Ek 5.3.1N</b> Documentation for an API/library is an important aspect of programming.</p> <p><b>Ek 5.3.1O</b> APIs connect software components, allowing them to communicate.</p>
<p><b>EU 5.4</b> Programs are developed, maintained, and used by people for different purposes.</p>	<p><b>IO 5.4.1</b> Evaluate the correctness of a program. [P4]</p>	<p><b>Ek 5.4.1A</b> Program style can affect the determination of program correctness.</p> <p><b>Ek 5.4.1B</b> Duplicated code can make it harder to reason about a program.</p> <p><b>Ek 5.4.1C</b> Meaningful names for variables and procedures help people better understand programs.</p> <p><b>Ek 5.4.1D</b> Longer code segments are harder to reason about than shorter code segments in a program.</p> <p><b>Ek 5.4.1E</b> Locating and correcting errors in a program is called debugging the program.</p> <p><b>Ek 5.4.1F</b> Knowledge of what a program is supposed to do is required in order to find most program errors.</p> <p><b>Ek 5.4.1G</b> Examples of intended behavior on specific inputs help people understand what a program is supposed to do.</p> <p><b>Ek 5.4.1H</b> Visual displays (or different modalities) of program state can help in finding errors.</p> <p><b>Ek 5.4.1I</b> Programmers justify and explain a program's correctness.</p> <p><b>Ek 5.4.1J</b> Justification can include a written explanation about how a program meets its specifications.</p> <p><b>Ek 5.4.1k</b> Correctness of a program depends on correctness of program components, including code segments and procedures.</p> <p><b>Ek 5.4.1l</b> An explanation of a program helps people understand the functionality and purpose of it.</p> <p><b>Ek 5.4.1m</b> The functionality of a program is often described by how a user interacts with it.</p> <p><b>Ek 5.4.1N</b> The functionality of a program is best described at a high level by what the program does, not at the lower level of how the program statements work to accomplish this.</p>
<p><b>EU 5.5</b> Programming uses mathematical and logical concepts.</p>	<p><b>IO 5.5.1</b> Employ appropriate mathematical and logical concepts in programming. [P1]</p>	<p><b>Ek 5.5.1A</b> Numbers and numerical concepts are fundamental to programming.</p> <p><b>Ek 5.5.1B</b> Integers may be constrained in the maximum and minimum values that can be represented in a program because of storage limitations.</p> <p><b>EXCIUSION STATEMENT (for Ek 5.5.1B):</b> Specific range limitations of all programming languages are beyond the scope of this course and the AP Exam.</p> <p><b>Ek 5.5.1C</b> Real numbers are approximated by floating-point representations that do not necessarily have infinite precision.</p> <p><b>EXCIUSION STATEMENT (for Ek 5.5.1C):</b> Specific sets of values that cannot be exactly represented by floating-point numbers are beyond the scope of this course and the AP Exam.</p> <p><b>Ek 5.5.1D</b> Mathematical expressions using arithmetic operators are part of most programming languages.</p> <p><b>Ek 5.5.1E</b> Logical concepts and Boolean algebra are fundamental to programming.</p> <p><b>Ek 5.5.1F</b> Compound expressions using and, or, and not are part of most programming languages.</p> <p><b>Ek 5.5.1G</b> Intuitive and formal reasoning about program components using Boolean concepts helps in developing correct programs.</p> <p><b>Ek 5.5.1H</b> Computational methods may use lists and collections to solve problems.</p> <p><b>Ek 5.5.1I</b> Lists and other collections can be treated as abstract data types (ADTs) in developing programs.</p> <p><b>Ek 5.5.1J</b> Basic operations on collections include adding elements, removing elements, iterating over all</p>



elements, and determining whether an element is in a collection.

## Big Idea 6: The Internet

### Essential Questions:

- ▶ What is the Internet? How is it built? How does it function?
- ▶ What aspects of the Internet's design and development have helped it scale and flourish?
- ▶ How is cybersecurity impacting the ever-increasing number of Internet users?

Enduring Understandings(Students will understand that ... )	Learning Objectives(Students will be able to ... )	Essential Knowledge (Students will know that ... )
<p><b>EU 6.1</b> The Internet is a network of autonomous systems.</p>	<p><b>IO 6.1.1</b> Explain the abstractions in the Internet and how the Internet functions. [P3]  <b>EXCIUSION STATEmENT (for IO 6.1.1):</b>Specific devices used to implement the abstractions in the Internet are beyond the scope of this course and the AP Exam.</p>	<p><b>Ek 6.1.1A</b> The Internet connects devices and networks all over the world.  <b>Ek 6.1.1B</b> An end-to-end architecture facilitates connecting new devices and networks on the Internet.  <b>Ek 6.1.1C</b> Devices and networks that make up the Internet are connected and communicate using addresses and protocols.  <b>Ek 6.1.1D</b> The Internet and the systems built on it facilitate collaboration.  <b>Ek 6.1.1E</b> Connecting new devices to the Internet is enabled by assignment of an Internet protocol (IP) address.  <b>Ek 6.1.1F</b> The Internet is built on evolving standards, including those for addresses and names.  <b>EXCIUSION STATEmENT (for Ek 6.1.1F):</b> Specific details of any particular standard for addresses are beyond the scope of this course and the AP Exam.  <b>Ek 6.1.1G</b> The domain name system (DNS) translates domain names to IP addresses.  <b>Ek 6.1.1H</b> The number of devices that could use an IP address has grown so fast that a new protocol (IPv6) has been established to handle routing of many more devices.  <b>Ek 6.1.1I</b> Standards such as hypertext transfer protocol (HTTP), IP, and simple mail transfer protocol (SMTP) are developed and overseen by the Internet EngineeringTask Force (IETF).</p>
<p><b>EU 6.2</b> Characteristics of the Internet influence the systems built on it.</p>	<p><b>IO 6.2.1</b> Explain characteristics of the Internet and the systems built on it. [P5]</p>	<p><b>Ek 6.2.1A</b> The Internet and the systems built on it are hierarchical and redundant.  <b>Ek 6.2.1B</b> The domain name syntax is hierarchical.<b>Ek 6.2.1C</b> IP addresses are hierarchical.  <b>Ek 6.2.1D</b> Routing on the Internet is fault tolerant and redundant.</p>
	<p><b>IO 6.2.2</b> Explain how the characteristics of the Internet influence the systems built on it. [P4]</p>	<p><b>Ek 6.2.2A</b> Hierarchy and redundancy help systems scale.  <b>Ek 6.2.2B</b> The redundancy of routing (i.e., more than one way to route data) between two points on the Internet increases the reliability of the Internet and helps it scale to more devices and more people.  <b>Ek 6.2.2C</b> Hierarchy in the DNS helps that system scale.<b>Ek 6.2.2D</b> Interfaces and protocols enable widespread use of the Internet.  <b>Ek 6.2.2E</b> Open standards fuel the growth of the Internet.  <b>Ek 6.2.2F</b> The Internet is a packet-switched system through which digital data is sent by breaking the data into blocks of bits called packets, which contain both the data being transmitted and control information for routing the data.  <b>EXCIUSION STATEmENT (for Ek 6.2.2F):</b> Specific details of any particular packet-switching system are</p>

		<p>beyond the scope of this course and the AP Exam.</p> <p><b>Ek 6.2.2G</b> Standards for packets and routing include transmission control protocol/Internet protocol (TCP/IP).</p> <p><b>EXCIUSION STATEMENT (for Ek 6.2.2G):</b> Specific technical details of howTCP/IP works are beyond the scope of this course and the AP Exam.</p> <p><b>Ek 6.2.2H</b> Standards for sharing information and communicating between browsers and servers on the Web include HTTP and secure sockets layer/transport layer security (SSL/TLS).</p> <p><b>EXCIUSION STATEMENT (for Ek 6.2.2H):</b> Understanding the technical aspects of how SSL/TLS works is beyond the scope of this course and the AP Exam.</p> <p><b>Ek 6.2.2I</b> The size and speed of systems affect their use.</p> <p><b>Ek 6.2.2J</b> The bandwidth of a system is a measure of bit rate — the amount of data (measured in bits) that can be sent in a fixed amount of time.</p> <p><b>Ek 6.2.2k</b> The latency of a system is the time elapsed between the transmission and the receipt of a request.</p>
<p><b>EU 6.3</b> Cybersecurity is an important concern for the Internet and the systems built on it.</p>	<p><b>IO 6.3.1</b> Identify existing cybersecurity concerns and potential options to address these issues with the Internet and the systems built on it. [P1]</p>	<p><b>Ek 6.3.1A</b> The trust model of the Internet involves trade-offs.</p> <p><b>Ek 6.3.1B</b> The DNS was not designed to be completely secure.</p> <p><b>Ek 6.3.1C</b> Implementing cybersecurity has software, hardware, and human components.</p> <p><b>Ek 6.3.1D</b> Cyberwarfare and cybercrime have widespread and potentially devastating effects.</p> <p><b>Ek 6.3.1E</b> Distributed denial-of-service attacks (DDoS) compromise a target by flooding it with requests from multiple systems.</p> <p><b>Ek 6.3.1F</b> Phishing, viruses, and other attacks have human and software components.</p> <p><b>Ek 6.3.1G</b> Antivirus software and firewalls can help prevent unauthorized access to private data.</p> <p><b>Ek 6.3.1H</b> Cryptography is essential to many models of cybersecurity.</p> <p><b>Ek 6.3.1I</b> Cryptography has a mathematical foundation.</p> <p><b>EXCIUSION STATEMENT (for Ek 6.3.1I):</b> Specific mathematical functions used in cryptography are beyond the scope of this course and the AP Exam.</p> <p><b>Ek 6.3.1J</b> Open standards help ensure cryptography is secure.</p> <p><b>Ek 6.3.1k</b> Symmetric encryption is a method of encryption involving one key for encryption and decryption.</p> <p><b>EXCIUSION STATEMENT (for Ek 6.3.1k):</b>The methods used in encryption are beyond the scope of this course and the AP Exam.</p> <p><b>Ek 6.3.1l</b> Public key encryption, which is not symmetric, is an encryption method that is widely used because of the functionality it provides.</p> <p><b>EXCIUSION STATEMENT (for Ek 6.3.1l):</b>The mathematical methods used in public key cryptography are beyond the scope of this course and the AP Exam.</p> <p><b>Ek 6.3.1m</b> Certificate authorities (CAs) issue digital certificates that validate the ownership of encrypted keys used in secured communications and are based on a trust model.</p> <p><b>EXCIUSION STATEMENT (for Ek 6.3.1m):</b>The technical details of the process CAs follow are beyond the scope of this course and the AP Exam.</p>

**Big Idea 7: Global Impact**

## Essential Questions:

- ▶ How does computing enhance human communication, interaction, and cognition?
- ▶ How does computing enable innovation?
- ▶ What are some potential beneficial and harmful effects of computing?
- ▶ How do economic, social, and cultural contexts influence innovation and the use of computing?

Enduring Understandings(Students will understand that ... )	Learning Objectives(Students will be able to ... )	Essential Knowledge (Students will know that ... )
<p><b>EU 7.1</b> Computing enhances communication, interaction, and cognition.</p>	<p><b>IO 7.1.1</b> Explain how computing innovations affect communication, interaction, and cognition. [P4]</p>	<p><b>EK 7.1.1A</b> Email, SMS, and chat have fostered new ways to communicate and collaborate.</p> <p><b>EK 7.1.1B</b> Video conferencing and video chat have fostered new ways to communicate and collaborate.</p> <p><b>EK 7.1.1C</b> Social media continues to evolve and fosters new ways to communicate.</p> <p><b>EXCLUSION STATEMENT (for Ek 7.1.1C):</b> Detailed knowledge of any social media site is beyond the scope of this course and the AP Exam.</p> <p><b>EK 7.1.1D</b> Cloud computing fosters new ways to communicate and collaborate.</p> <p><b>EK 7.1.1E</b> Widespread access to information facilitates the identification of problems, development of solutions, and dissemination of results.</p> <p><b>EK 7.1.1F</b> Public data provides widespread access and enables solutions to identified problems.</p> <p><b>EK 7.1.1G</b> Search trends are predictors.</p> <p><b>EK 7.1.1H</b> Social media, such as blogs and Twitter, have enhanced dissemination.</p> <p><b>EK 7.1.1I</b> Global Positioning System (GPS) and related technologies have changed how humans travel, navigate, and find information related to geolocation.</p> <p><b>EK 7.1.1J</b> Sensor networks facilitate new ways of interacting with the environment and with physical systems.</p> <p><b>EK 7.1.1k</b> Smart grids, smart buildings, and smart transportation are changing and facilitating human capabilities.</p> <p><b>EK 7.1.1l</b> Computing contributes to many assistive technologies that enhance human capabilities.</p> <p><b>EK 7.1.1m</b> The Internet and the Web have enhanced methods of and opportunities for communication and collaboration.</p> <p><b>EK 7.1.1N</b> The Internet and the Web have changed many areas, including e-commerce, health care, access to information and entertainment, and online learning.</p> <p><b>EK 7.1.1O</b> The Internet and the Web have impacted productivity, positively and negatively, in many areas.</p>
	<p><b>IO 7.1.2</b> Explain how people participate in a problem- solving process that scales. [P4]</p>	<p><b>EK 7.1.2A</b> Distributed solutions must scale to solve some problems.</p> <p><b>EK 7.1.2B</b> Science has been impacted by using scale and "citizen science" to solve scientific problems using home computers in scientific research.</p> <p><b>EK 7.1.2C</b> Human computation harnesses contributions from many humans to solve problems related to digital data and the Web.</p> <p><b>EK 7.1.2D</b> Human capabilities are enhanced by digitally enabled collaboration.</p> <p><b>EK 7.1.2E</b> Some online services use the contributions of many people to benefit both individuals and society.</p> <p><b>EK 7.1.2F</b> Crowdsourcing offers new models for collaboration, such as connecting people with jobs and businesses with funding.</p> <p><b>EK 7.1.2G</b> The move from desktop computers to a proliferation of always-on mobile computers is leading</p>

		to new applications.
<b>EU 7.2</b> Computing enables innovation in nearly every field	<b>IO 7.2.1</b> Explain how computing has impacted innovations in other fields. [P1]	<p><b>Ek 7.2.1A</b> Machine learning and data mining have enabled innovation in medicine, business, and science.</p> <p><b>Ek 7.2.1B</b> Scientific computing has enabled innovation in science and business.</p> <p><b>Ek 7.2.1C</b> Computing enables innovation by providing the ability to access and share information.</p> <p><b>Ek 7.2.1D</b> Open access and Creative Commons have enabled broad access to digital information.</p> <p><b>Ek 7.2.1E</b> Open and curated scientific databases have benefited scientific researchers.</p> <p><b>Ek 7.2.1F</b> Moore's law has encouraged industries that use computers to effectively plan future research and development based on anticipated increases in computing power.</p> <p><b>Ek 7.2.1G</b> Advances in computing as an enabling technology have generated and increased the creativity in other fields.</p>
<b>EU 7.3</b> Computing has global effects — both beneficial and harmful — on people and society.	<b>IO 7.3.1</b> Analyze the beneficial and harmful effects of computing. [P4]	<p><b>Ek 7.3.1A</b> Innovations enabled by computing raise legal and ethical concerns.</p> <p><b>Ek 7.3.1B</b> Commercial access to music and movie downloads and streaming raises legal and ethical concerns.</p> <p><b>Ek 7.3.1C</b> Access to digital content via peer-to-peer networks raises legal and ethical concerns.</p> <p><b>Ek 7.3.1D</b> Both authenticated and anonymous access to digital information raise legal and ethical concerns.</p> <p><b>Ek 7.3.1E</b> Commercial and governmental censorship of digital information raise legal and ethical concerns.</p> <p><b>Ek 7.3.1F</b> Open source and licensing of software and content raise legal and ethical concerns.</p> <p><b>Ek 7.3.1G</b> Privacy and security concerns arise in the development and use of computational systems and artifacts.</p> <p><b>Ek 7.3.1H</b> Aggregation of information, such as geolocation, cookies, and browsing history, raises privacy and security concerns.</p> <p><b>Ek 7.3.1I</b> Anonymity in online interactions can be enabled through the use of online anonymity software and proxy servers.</p> <p><b>Ek 7.3.1J</b> Technology enables the collection, use, and exploitation of information about, by, and for individuals, groups, and institutions.</p> <p><b>Ek 7.3.1k</b> People can have instant access to vast amounts of information online; accessing this information can enable the collection of both individual and aggregate data that can be used and collected.</p> <p><b>Ek 7.3.1l</b> Commercial and governmental curation of information may be exploited if privacy and other protections are ignored.</p> <p><b>Ek 7.3.1m</b> Targeted advertising is used to help individuals, but it can be misused at both individual and aggregate levels.</p> <p><b>Ek 7.3.1N</b> Widespread access to digitized information raises questions about intellectual property.</p> <p><b>Ek 7.3.1O</b> Creation of digital audio, video, and textual content by combining existing content has been impacted by copyright concerns.</p> <p><b>Ek 7.3.1P</b> The Digital Millennium Copyright Act (DMCA) has been a benefit and a challenge in making copyrighted digital material widely available.</p> <p><b>Ek 7.3.1Q</b> Open source and free software have practical, business, and ethical impacts on widespread access to programs, libraries, and code.</p>
<b>EU 7.4</b> Computing innovations influence and are influenced by the economic, social, and cultural contexts in which they are designed and used.	<b>IO 7.4.1</b> Explain the connections between computing and real-world contexts, including economic, social, and cultural contexts. [P1]	<b>Ek 7.4.1A</b> The innovation and impact of social media and online access varies in different countries and in different socioeconomic groups.

		<p><b>Ek 7.4.1B</b> Mobile, wireless, and networked computing have an impact on innovation throughout the world.</p> <p><b>Ek 7.4.1C</b> The global distribution of computing resources raises issues of equity, access, and power.</p> <p><b>Ek 7.4.1D</b> Groups and individuals are affected by the "digital divide" — differing access to computing and the Internet based on socioeconomic or geographic characteristics.</p> <p><b>Ek 7.4.1E</b> Networks and infrastructure are supported by both commercial and governmental initiatives.</p>
<p><b>EU 7.5</b> An investigative process is aided by effective organization and selection of resources. Appropriate technologies and tools facilitate the accessing of information and enable the ability to evaluate the credibility of sources.</p>	<p><b>IO 7.5.1</b> Access, manage, and attribute information using effective strategies. [P1]</p>	<p><b>Ek 7.5.1A</b> Online databases and libraries catalog and house secondary and some primary sources.</p> <p><b>Ek 7.5.1B</b> Advance search tools, Boolean logic, and key words can refine the search focus and/or limit search results based on a variety of factors (e.g., data, peer-review status, type of publication).</p> <p><b>Ek 7.5.1C</b> Plagiarism is a serious offense that occurs when a person presents another's ideas or words as his or her own. Plagiarism may be avoided by accurately acknowledging sources.</p>
	<p><b>IO 7.5.2</b> Evaluate online and print sources for appropriateness and credibility. [P5]</p>	<p><b>Ek 7.5.2A</b> Determining the credibility of a source requires considering and evaluating the reputation and credentials of the author(s), publisher(s), site owner(s), and/or sponsor(s).</p> <p><b>Ek 7.5.2B</b> Information from a source is considered relevant when it supports an appropriate claim or the purpose of the investigation.</p>

**Required Curriculum Materials:** (This section should contain information regarding textbooks, technology integration, films, videos and various resources used in teaching the course. Please note whether items already exist in-district or will need to be purchased. Any additional notes that are useful to teachers should be included.)

Students need access computers and wifi. Beyond that, all other curricular materials are included are general classroom supplies.