AP® Computer Science Principles

Course Overview

AP® Computer Science Principles can be taught as a single course or in conjunction with a pre-ap course. AP® Computer Science Principles will cover a wide range of Computer Science concepts. The programming language used to support the concepts covered is completely up to teacher discretion. The course will also cover concepts that are not focused solely on programming like the Internet and global impact of computing on society. Students will research topics in Computer Science, read relevant Computer Science related articles, and write responses and reports about the research and articles.

AP® Computer Science Principles should be taught in a computer lab where each student has access to a computer. The best configuration is to have all PCs set against the outer wall of the room where the instructor can view all computer screens easily at all times. The students should work on programming assignments individually and collaboratively as the particular assignment dictates.

Each student in each class must have access to a computer during class, before school, and after school. For each unit of coverage that involves programming, there should be multiple computer-based programming lab assignments.

Students perform at the highest level when differentiation is used and they are given many differing assignment options from which to choose. The more options the students can choose from the higher the rate of student success, considering that all students do not learn the same way nor do they have the same interests.

During new topic discussions involving programming concepts, students should open sample programs on the computer to see the topics in code form. Students should be instructed to make changes to the sample programs and then run the programs to test the changes so that each student can gain some practical experience working with code. Students should have many opportunities to use computers.

The Create Performance Task can be completed in a single unit or as pieces of multiple units. This syllabus has built in sections for completion of the performance tasks. It is not assumed that the performance task must be completed in only one unit. The performance task could be started earlier in the semester and then completed later. The Create Performance Task requires a minimum of 12 hours of class time for completion.

Curricular Requirements

CR1 The teacher and students have access to college-level computer science resources, in print or electronic format. See page: 3

CR2 The course provides opportunities to develop student understanding of the required content outlined in each of the big ideas described in the AP Course and Exam Description (CED). See page: pages 4-19

CR3 The course provides opportunities to develop student understanding of the big ideas, as outlined in the AP Course and Exam Description. See page: pages 4-19

CR4 The course provides opportunities for students to develop the skills related to Computational Thinking Practice 1: Computational Solution Design. See page: 4,5,12,14,15,and 17

CR5 The course provides opportunities for students to develop the skills related to Computational Thinking Practice 2: Algorithms and Program Development. See page: 8,9,10,13,14,15, and 16

CR6 The course provides opportunities for students to develop the skills related to Computational Thinking Practice 3: Abstraction in Program Development. See page: 6,7,9,10,14,15, and 16

CR7 The course provides opportunities for students to develop the skills related to Computational Thinking Practice 4: Code Analysis. See page: 8,9,10,12,13,14,15,16, and 17

CR8 The course provides opportunities for students to develop the skills related to Computational Thinking Practice 5: Computing Innovations. See page: 4,6,7,17,18, and 19

CR9 The course provides opportunities for students to develop the skills related to Computational Thinking Practice 6: Responsible Computing. See page: 4,17,18, and 19

CR10 The course provides a minimum of three opportunities for students to investigate different computing innovations. See page: 4,6,17,18, and 19

CR11 Students are provided at least 12 hours of dedicated class time to complete the AP Create Performance Task. See page: pages 4-19

Resource Requirements

The school ensures that each student has access to a computer with Internet access to prepare for and to complete the AP Computer Science Principles Performance Task in class.

The computer must be able to access the Internet sites necessary for students to be successful in the course and assessment.

The school ensures that the computer system(s) available for students contains appropriate software to create and edit programs and other computational artifacts, and to allow students to practice for and to complete the AP Computer Science Principles Performance Task.

The school ensures that each student has access to the AP Computer Science Principles Exam Reference Sheet, and Performance Task and Performance Task Rubrics.

The school ensures that each student has a college level text or curricular resources for individual use inside and outside of the classroom.

Resources

Abelson, Ledeen, and Lewis. *Blown to Bits [**Free PDF**].* 2015. http://www.bitsbook.com/

Armstrong, Stacey. *A+ Computer Science: Computer Science Curriculum Solutions.* http://apluscompsci.com, 2020

Cisco Binary Game. https://learningcontent.cisco.com/games/binary/index.html

Scratch. MIT. https://scratch.mit.edu/

Snap. Berkeley. http://snap.berkeley.edu/

Python. https://www.python.org/

Lots of current online articles that demonstrate the impact of Computer Science on the world around us.

TIME	TOPICS
1 week	**Unit 1 –** What is Computer Science? **Topics -** Impacts of computer science on society; privacy; legal issues and intellectual property; social and ethical ramifications of computer use. **Student Objectives and Activities -** Students will learn what Computer Science is and how it affects the world around us. Students will explain how computing innovations affect other fields and analyze the beneficial and harmful effects of computing. Students will explain the connections between computing and economic, social, and cultural contexts. **(IOC, P5, P6)** Students will work in groups to research the impact computing and computer science has had on society, innovation, and the economy. The impact can be positive or negative. The groups will use sources, including the Internet to gather information. The groups will create an artifact that highlights the group's research. **(CRD, IOC, P1, P5, P6)** Groups will read Blown to Bits chapters as part of their research process. Students will write responses and / or create presentations after reviewing relevant computer science articles, evaluating computer artifacts, and conducting online research. **(CRD, IOC, P5, P6)** Students will learn the proper approach to using online resources as well as how to acknowledge the works of others that are posted on the Internet. **(IOC, P5, P6)** **[Skill 1.C, Skill 5.C, Skill 6.A, Skill 6.B]** *Guided Practice : Online research* *Readings : Online resources, online articles, Blown to Bits* *Assessments : Evaluation of document created from research.* **Big Ideas –** Creativity Development and Impact of Computing [CRD, IOC] **Practices –** Computational Solution Design, Computing Innovations, and Responsible Computing [P1, P5, P6]

1 week	**Unit 2 –** What is a program? **Topics -** Program Development; Input; Output; Code statements; Code segments. **Student Objectives and Activities -** Students will learn why programs are created, approaches to build a program, and the components needed to build a program. **(P1)** Students will discuss reasons for and the process of creating a program. **(CRD, P1)** Students will create programs that read in and print out values. **(CRD, P1)** Students will learn about errors and how to handle them. **(CRD, P1)** **[Skill 1.A, Skill 1.B]** *Guided Practice : Online research* *Readings : Online resources, online articles, Blown to Bits* *Assessments : Quiz, Test, Written Responses* **Big Ideas –** Creativity Development [CRD] **Practices –** Computational Solution Design [P1]

1 week	**Unit 3 –** Abstraction **Topics –** Bits; binary numbers; base conversion; languages; data types; compression **Student Objectives and Activities –** Students will learn about the different levels of abstraction. **(DAT, P3)** Students will learn about binary numbers. Students will learn that programming languages are higher levels of abstraction than binary. Students will describe the variety of abstractions used to represent data. Students will explain how binary sequences are used to represent digital data. **(DAT, P3)** Students will demonstrate knowledge of number bases and converting between bases by completing a worksheet and playing number system games. Students will compare and contrast low level binary with high level programming code. **(DAT, P3)** Students will compare and contrast Lossy and Lossless compression. **(DAT, P3)** Students will read Blown to Bits chapters. Students will write responses and / or create a presentation at the end of the unit. **(P3, P5)** **[Skill 3.C, Skill 5.A]** *Guided Practice : Topic discussion, Worksheets* *Guided Practice : Cisco Binary Game* *https://learningnetwork.cisco.com/s/binary-game* *Readings : Slides, Worksheets, Online Articles, Blown to Bits* *Readings : CS Unplugged Binary Activities* *Assessments : Quiz, Test, Written Responses* **Big Ideas –**Data [DAT] **Practices –**Abstraction in Program Development and Computing Innovations [P3, P5]

1 week	**Unit 4 –** Data Part 1 **Topics -** Data, Variables **Student Objectives and Activities –** Students will learn how data is stored. Students will learn about the costs and risks of storing large sets of data. Students will learn what must be done to large sets of data in order to gather information. **(DAT, IOC, P5)** Students will research articles on big data storage and big data tools to learn about the innovations related to big data storage and processing. **(DAT, IOC, P5)** Student will learn how to create and use a variable to store values. **(DAT, P3)** Students will create a program that uses variables to store values. Students will manipulate the values stored in a variable and print the results. **(DAT, P3)** **[Skill 3.A, Skill 5.B, Skill 5.D]** *Guided Practice : Online research* *Readings : Online resources, online articles, Blown to Bits* *Assessments : Evaluation of document created from research.* *Assessments : Quiz, Test, Written Responses* **Big Ideas –** Data, Impact of Computing [DAT, IOC] **Practices –** Abstraction in Program Development and Computing Innovations [P3, P5]

3 weeks	**Unit 5 –** Creating digital artifacts with code – Iteration and Selection **Topics –** Variables, conditional statements, and loop. **Student Objectives and Activities –** Students learn how to use if, if else, and if else if statements to test conditions and add decision making to their programs using boolean conditions and variables. Students learn about data type limitations and appropriate use of data types. Students learn how to use loops and how to add iterative processes to their programs. Students will examine code segments that use variables, selection, and iteration. **(CRD, P2, P4)** Students will collaborate to design a game that uses variables to keep score. **(CRD, DAT, P2)** Students will use a visual programming environment to create programs that use variables, decision-making, and iteration. Students will create a program that uses conditional statements to determine how many times a sprite bounces off the wall. **(CRD, DAT, P2)** Students will create programs using loops that draw geometrical shapes. **(CRD, P2)** Students will use loops to design and create simple games. Students will incorporate key presses and input to make the programs more dynamic. **(CRD, P2, P4)** Students will work with a group to create a game that uses variables, conditionals, loops, and input / key presses; such as ping pong. **(CRD, DAT, P2, P4)** **[Skill 2.B, Skill 4.A, Skill 4.B]** *Guided Practice : Topic discussion, Example program analysis and modification, Worksheets* *Readings : Labs, Slides, Worksheets, etc.* *Assessments : Labs, Quizzes, and Test* **Big Ideas –** Creativity Development, Data [CRD, DAT] **Practices –** Algorithms and Program Development and Code Analysis [P2, P4]

3 weeks	**Unit 6 –** List Processing **Topics –** Lists; Traversals; Insertions; Deletions; **Student Objectives and Activities –** Students will learn how to use a list, add items to a list, organize items in a list, and delete items from a list. Students will design programs that use lists to create animations and games. **(CRD, DAT, AAP, P2, P3, P4)** Students will use a visual programming **environment** to write programs that iterate through lists to count the number of odds and evens. Students will master iterating through a list with a loop using list indexes to access each individual value in the list. **(CRD, DAT, AAP, P2, P3, P4)** Students will learn to debug list processing code. **(P2, P4)** Students will create a program to count the number of occurrences of a value in a list. **(DAT, AAP, P2)** Students will create a program to create a list of factors of a number. **(DAT, AAP, P2)** Students will work in pairs or groups to create a simple game that uses lists. **(CRD, DAT, AAP)** **[Skill 2.A, Skill 2.B, Skill 3.A, Skill 4.A, Skill 4.B]** *Guided Practice : Topic discussion, Example program analysis and modification, Worksheets* *Readings : Labs, Slides, Worksheets, etc.* *Assessments : Labs, Quizzes, and Tests* **Big Ideas –** Creativity Development, Data, Algorithms and Programming [CRD, DAT, AAP] **Practices –**Algorithms and Program Development, Abstraction in Program Development and Code Analysis [P2, P3, P4]

4 weeks	**Unit 7 –** Algorithms and Abstraction **Topics –** Lists; Abstraction; Blocks **Student Objectives and Activities –** Students will learn several different algorithms; find the smallest item, largest item, item that occurs the most, item that occurs the least, and / or a sort or search or both. **(CRD, AAP, P2)** Students will evaluate pseudocode algorithms. Students will learn how a non-programming language algorithm can be converted to a specific language algorithm. **(AAP, P4)** Students will develop an abstraction, use multiple levels of abstraction, and identify multiple levels of abstractions that are used when writing programs or creating other computational artifacts. **(CRD, AAP, P2, P3)** Students will use a visual programming environment to create a program with a block that determines if the parameter passed in is odd or even. Students will write a program with two blocks – one block will check for odd or even and the second block will iterate through a list to count the number of odd or even values. **(CRD, AAP, P3, P4)** Students will identify the levels of abstraction used in a program that uses blocks to solve a problem. Students will demonstrate mastery of calling a block and passing in values. **(CRD, AAP, P2, P3)** Students will debug code with blocks and parameter passing. **(CRD, AAP, P2, P3)** Student groups will expand the ping pong game to include blocks and abstraction. **(CRD)** **[Skill 2.A, Skill 2.B, Skill 3.A, Skill 3.C, Skill 4.A, Skill 4.B]** *Guided Practice : Topic discussion, Example program analysis and modification, Worksheets* *Readings : Labs, Slides, Worksheets, etc.* *Assessments : Labs, Quizzes, and Tests* **Big Ideas –** Creativity Development, Algorithms and Programming [CRD, AAP] **Practices –**Algorithms and Program Development, Abstraction in Program Development and Code Analysis [P2, P3, P4]

4 weeks	**Unit 8 –** Semester Projects and Performance Task **Topics –** Collaboratively work to create an simulation, animation, or game. Real world topics should be researched and used a back drop for this project. Work on the Performance Task. **Student Objectives and Activities –** Students will work with a partner to design a simulation, animation, or game. Each group will create a computational artifact using computing tools and techniques to solve a problem. Students will use the knowledge gained from the group project to help when they individually complete the Performance Task. Students will work with a partner to design a simulation, animation, or game – dna simulation, molecular simulation, or arcade game. Each member of the group will design components and then work together to assemble the components. Each group will explain how they assembled the artifact and how that process worked. Students will work individually on the Performance Task. **[Many skills are incorporated for this unit.]** *Guided Practice : Topic discussion, Example program analysis and modification, Worksheets* *Readings : Labs, Slides, Worksheets, online articles, Blown to Bits* *Assessments : Evaluation of computational artifact and / or written responses* *Performance Task - Students spend at least 12 hours creating a digital artifact.* **Big Ideas –** Most are incorporated in this unit. **Practices –** Most are incorporated in this unit.
	END OF SEMESTER ONE

TIME	TOPICS
1 week	**Unit 9 –** What is a program part 2? **Topics -** Program Development; Input; Output; Code statements; Code segments. **Student Objectives and Activities –** Students will learn the differences between pseudocode, block- based languages, and text-based languages. **(P1, P4)** Students will learn the difference between syntax, runtime, and logic errors. **(P1, P4)** Students will discuss reasons to create a program and the process of creating a program. **(P4)** Students will create basic text-based programs that read in values and print out values. **(CRD, P1, P4)** Students will learn about errors and how to handle them. **(CRD, P4)** **[Skill 1.A, Skill 4.C]** *Guided Practice : Online research* *Readings : Online resources, online articles, Blown to Bits* *Assessments : Quiz, Test, Written Responses* **Big Ideas –** Creativity Development [CRD] **Practices –** Computational Solution Design and Code Analysis [P1, P4]

2 weeks	**Unit 10 –** Variables and Decision Making **Topics –** Variables and Conditional Statements. **Student Objectives and Activities –** Students learn how to use if, if else, and if else if statements to test conditions and add decision making to their programs using boolean conditions and variables. **(CRD, P4)** Students will use a text based programming language to create programs that use variables and conditional statements. **(CRD, DAT, AAP, P2)** Students will compare and contrast block-based and text-based variable creation. **(DAT, P4)** Students will debug code and learn to develop code from algorithms. **(DAT, AAP, P2, P4)** Students will create a program that uses conditional statements to make a simple image move around the screen. **(CRD, AAP, P2)** Students will work with a group to create a basic game that uses variables and conditionals. Classes can be used as an optional extension. **(CRD, DAT, AAP, P2, P4)** **[Skill 2.B, Skill 4.A]** *Guided Practice : Topic discussion, Example program analysis and modification, Worksheets* *Readings : Labs, Slides, Worksheets, etc.* *Labs : odd or even, discount calculator* *Assessments : Labs, Quizzes, and Test* **Big Ideas –** Creativity Development, Data, Algorithms and Programming [CRD, DAT, AAP] **Practices –** Algorithms and Program Development and Code Analysis [P2, P4]

2 weeks	**Unit 11 –** Abstraction and Iteration **Topics –** Abstraction, iteration, parameters, and method calling **Student Objectives and Activities –** Students will use methods, create methods, pass parameters, and learn about abstraction. **(P3)** Students learn how to use loops and how to add iterative processes to their programs. Students will create basic programs and examine example programs. **(AAP, P4)** Students will combine Boolean conditions and variables learned in prior units with looping concepts learned in this unit. Students learn the different parts of a loop and when to use a particular type of loop. **(CRD, DAT, AAP, P1, P2, P3)** Students will use graphics to draw any shape they choose; such as a smiley face or a car. Students will choose a shape to draw, design an algorithm to create the shape, turn that algorithm into a program, and build the program using drawing methods. **(CRD, P1)** Students will create an animation by adding iteration to the shape program to make the shape move. Students will take the shape draw in the previous unit and add iteration to make the shape bounce around the screen. Students will create methods and pass parameters, learning more about abstraction. **(CRD, AAP, P1, P4)** Students will design and create a program that uses loops to sum all of the digits in a number. Students will use graphics and loops to draw a series of decreasing circles and squares. Students will start building an arcade game working in pairs. **(DAT, AAP, P2, P3)** **[Skill 1.A, Skill 2.A, Skill 2.B, Skill 3.A, Skill 3.B, Skill 4.A, Skill 4.B]** *Guided Practice : Topic discussion, Example program analysis and modification, Worksheets* *Readings : Labs, Slides, Worksheets, etc.* *Assessments : Evaluation of artifacts created* **Big Ideas –** Creativity Development, Data, and Algorithms and Programming [CRD, DAT, AAP] **Practices –** Computational Solution Design, Algorithms and Program Development, Abstraction in program Development, and Code Analysis [P1, P2, P3, P4]

3 weeks	**Unit 12 –** Processing Strings and Lists **Topics –** Lists; Traversals; Insertions; Deletions; Abstraction; Algorithms **Student Objectives and Activities –** Students will learn how to use a String. Students will learn how to access the values in a String. Students will examine and write simple programs using Strings. **(DAT, AAP, P1, P4)** Students will learn how to use a list, add items to a list, organize items in a list, and delete items from a list. Students will examine and write simple programs using lists. **(DAT, AAP, P1, P4)** Students will explain the difference between list processing algorithms. Students will debug programs that use loops to iterate through list indexes. Students will create programs to sum all values in a list and to count all evens and odds in a list. **(CRD, DAT, AAP, P1, P2, P3)** Students will work in pairs to create programs and to fix broken programs. **(P4)** Students will write a program that will use a list of images to create a sprite animation. Students will use loops and lists to create a program that creates a sliding background. Students will work in groups to combine the sprite animation and background slider to begin creating an arcade game. **(CRD, P1)** **[Skill 1.A, Skill 2.A, Skill 2.B, Skill 3.A, Skill 4.A, Skill 4.B]** *Guided Practice : Topic discussion, Example program analysis and modification, Worksheets* *Readings : Labs, Slides, Worksheets, etc.* *Assessments : Labs, Quizzes, and Test* **Big Ideas –** Creativity Development, Data, Algorithms and Programming [CRD, DAT, AAP] **Practices –** Computational Solution Design, Algorithms and Program Development, Abstraction in Program Development, and Code Analysis [P1, P2, P3, P4]

3 weeks	**Unit 13 –** Developing Algorithms **Topics –** Algorithm development, analysis, efficiency, and refinement **Student Objectives and Activities –** Students will learn how to design an algorithm, turn an algorithm into a program, repair an algorithm, and why certain algorithms work better than others in certain situations. Students will learn how some algorithms are more efficient than others. **(P2, P3, P4, P5)** Students will learn several different algorithms; find the smallest item, largest item, item that occurs the most, item that occurs the least, and / or a sort or search. **(CRD, DAT, AAP)** Students will learn about algorithmic efficiency. Students will learn that some algorithms will run in reasonable time and some will not. There are some problems for which an algorithm will not run in reasonable time. **(AAP, P4)** Students will learn how to convert a pseudocode [non-programming language specific] algorithm into high level program code. **(AAP, P2, P3, P4, P5)** **[Skill 2.A, Skill 2.B, Skill 3.A, Skill 3.B, Skill 3.C, Skill 4.A, Skill 4.B]** *Guided Practice : Topic discussion, Example program analysis and modification, Worksheets* *Readings : Labs, Slides, Worksheets, Online Articles* *Assessments : Labs, Quizzes, and a Test* **Big Ideas –** Creativity Development, Data, Algorithms and Programming [CRD, DAT, AAP] **Practices –** Algorithms and Program Development, Abstraction in Program Development, and Code Analysis [P2, P3, P4]

1 week	**Unit 14 –** Looking for patterns in Big Data **Topics –** Big data; algorithms; **Student Objectives and Activities –** Students will discuss how computers process information and find patterns. Students will search online for advancements and innovations that have been created to help with processing data. **(CSN, P5)** Students will analyze algorithms that search for values. Students will work in pairs or groups to compare large data files and small data files to see which algorithms work best. **(P4)** Students will work in pairs to compare and contrast working with large data sets in a text-based programming language and in a spread sheet application. **(DAT)** Students will research how and where large data sets are posted online. Students will discuss the risks of storing large sets of data. Students will research large data sets and download a large data set that they find interesting. **(IOC)** Students will draw conclusions about the big data set and print out diagrams or charts to back up their findings. Is it easier to process a big data set using a programming language or using a spread sheet? Groups will read Blown to Bits chapters as part of their research process. Students will report their conclusions in a class discussion. **(P1 and P6)** **[Skill 1.A, Skill 4.A, Skill 4.B, Skill 5.D, and Skill 6.A]** *Guided Practice : Topic discussion, Example program analysis and modification, Worksheets* *Readings : Labs, Slides, Worksheets, online articles* *Readings : Blown to Bits* *Resources : Online large data set repositories - Baby Names, Songs, Data.gov* *http://www.datasciencecentral.com/profiles/blogs/big-data-sets-available-for-free* **Big Ideas –** Data, Computer Systems and Networks, and Impact of Computing [DAT, CSN, and IOC] **Practices –** Computational Solution Design, Code Analysis, Computing Innovations, and Responsible Computing [P1, P4, P5, and P6]

1 week	**Unit 15 –** Computing Systems and Networks **Topics –** Discuss the Internet; Research the Internet; HMTL Basics **Student Objectives and Activities –** Students will learn more about the Internet and basic html. **(CSN, P5, P6)** Students will learn how a computer works, hardware, software, basic computer operations, integrity, and responsible use of the computer. **(CSN)** Students will learn some simple html to use to create a web page that contains a link, an image, and several different fonts and colors. Simple html pages will be provided as examples. **(CD, CSN)** Students will research the internet and how it was developed. Students will explain the abstractions in the Internet and how the Internet functions, the characteristics of the Internet and the systems built on it, how the characteristics of the Internet influence the systems built on it. **(CSN, P5 and P6)** If possible, students will get the opportunity to work with old computer hardware components to better understand how an actual computer is assembled. **(CSN, P5)** Groups will read Blown to Bits appendix chapter over the Internet as part of their research process. Students will create a digital artifact that includes specific internet explanations and diagrams as well as cyber security research. **(CSN, P6)** **[Skill 5.A, Skill 5.C, Skill 5.E, Skill 6.A]** *Guided Practice : Topic discussion and worksheets* *Readings : slides, worksheets, online articles, Blown to Bits Appendix – The Internet* *Activities – CS Unplugged* *Assessments : M/C Test* **Big Ideas –** Creativity Development and Computer Systems and Networks [CD, CSN] **Practices –** Computing Innovations and Responsible Computing [P5 and P6]

1 week	**Unit 16 –** Impact of Computing **Topics –** Discuss the Internet; Research Innovations and Advancements related to Computer Science; Impact of Computing **Student Objectives and Activities –** Students will explain how computing innovations affect communication, interaction, and cognition, how people participate in a problem-solving process that scales, how computing has impacted innovations in other fields, and analyze the beneficial and harmful effects of computing. Students will explain the connections between computing and economic, social, and cultural contexts. **(IOC , P5)** Students will identify existing cyber security concerns and potential options to address these issues with the Internet and the system built on it. **(IOC, P5 and P6)** Students will discuss the Digital Divide and Crowdsourcing. **(IOC, P5 and P6)** Students will explain how use the Internet safely. **(IOC, P6)** Groups will read Blown to Bits appendix chapter over the Internet as part of their research process. Students will create a digital artifact that includes specific internet explanations and diagrams as well as cyber security research. **(CSN, P6)** **[Skill 5.A, Skill 5.C, Skill 5.E, Skill 6.A]** *Guided Practice : Topic discussion and worksheets* *Readings : slides, worksheets, online articles, Blown to Bits* *Assessments : M/C Test* **Big Ideas –** Creativity Development, Computer Systems and Networks, and Impact of Computing [CD, CSN, IOC] **Practices –** Computing Innovations and Responsible Computing [P5 and P6]

4 weeks	**AP Review Time / Performance Task** *Guided Practice : Past year's multiple choice questions* *Guided Practice : slides and examples* *Guided Practice : Finalize the Performance Task if needed.* *Readings : Past multiple choice questions* *Readings : Review book units* *Readings : Blown to Bits chapters* ***Computational Thinking Practices*** *1. Computational Solution Design* *2. Algorithms and Program Development* *3. Abstraction in Program Development* *4. Code Analysis* *5. Computing Innovations* *6. Responsible Computing* ***Big Ideas*** *1. Creativity Development* *2. Data* *3. Algorithms and Programming* *4. Computer Systems and Networks* *5. Impact of Computing*
	End of Semester Two

Teaching / Evaluation Strategies

Topics are broken down into manageable pieces. Each topic is introduced and discussed in a group setting.

Sections of each topic that are programming specific are discussed and then reinforced using example computer programs. Students run the example programs, make changes to the programs, and ask questions about the programs. Students are presented with many examples and explanations for all topics presented.

Online articles relevant to Computer Science will be used throughout. Online research of relevant Computer Science topics will also be used. Students will write response to prompts and chosen topics throughout the course.

Blown to Bit is a book that explains many relevant Computer Science topics. It can be used in the course where appropriate. The PDF version of the book is a free download.

Worksheets are provided that enhance the discussions and provide students the opportunity to practice the concepts without having to use the computer.

Lab time is provided so that each student has the opportunity to apply the programming concepts in a hands-on situation using a PC. Lab time is also provided to work on written responses.

For each programming topic, there are many computer-based programming lab assignments so that each student has the opportunity to practice the topic in different ways. Pair programming will be utilized on many of the projects. Games and simulations will be used as projects and groups will be utilized for some of these projects.

Quizzes are given to provide feedback and to gain information about the learning process. Tests are given in multiple-choice and free response format in a way that models what students will see on the AP test.

The Performance Task can be completed in a single unit or as pieces of multiple units. It is not assumed that the performance tasks must be completed in only one unit. The Performance Task requires a minimum of 12 hours of class time for completion.