AP® Computer Science A

Course Overview

The A+ curriculum for the AP Computer Science A course provides thorough coverage of all Big Ideas, Computational Thinking Practices, Learning Objectives, and Essential Knowledge and Skills as described in the *AP Computer Science A Course and Exam Description*. All teachers teaching AP Computer Science AP A should be very familiar with all of the detailed information provided in the current course and exam description.

This AP CS A Syllabus includes reference to the AP CS A optional labs – Magpie, Elevens, Picture, Consumer, Data, Stenography, and Celebrity. These or equivalent labs should be used early and often to help reinforce the core concepts tested on the AP Computer Science A exam and to provide students with at least 20 hours of programming time which is one of the curriculum requirements for the AP CS A course.

Teacher Resources

Armstrong, Stacey. *A+ Computer Science: Computer Science Curriculum Solutions.* http://apluscompsci.com, 2021

The College Board. *AP CS A Labs*. New York: College Entrance Examination Board, 2021.

Parlante, Nick. CodingBat. www.codingbat.com. Stanford University, 2021.

Teukolsky, Roselyn. *Barron's AP Computer Science A, 9th* ed. Hauppauge, N.Y.: Barron's Educational Series, 2019.

"Blue Pelican Java," by Charles E. Cook. ISBN 1-58939-758-4

TIME	TOPICS
2 weeks	**Computer Science / Programming Fundamentals / Java Basics** **Course Topics [CED Unit Topics 1.1 - 1.5 and 2.8]** Input and Output Primitive Types Variables and Data Types Math operations Assignment Statements and Expressions Methods and Parameters Social and Ethical Implications of Computer Use **Student Objectives** Students will discuss why the Java language is used and where. Students will learn how computers work and how programs are used to provide instructions to computers. Students will print values to the output window and read in values from an input device. Students will use string literals in print statements. (VAR) Students will learn about primitive types, including int and double. (VAR) Students will learn about different data types, which types can store which values, and how to calculate values and store the values. (VAR) Students will assign values to variables and create expressions to calculate values. (VAR) Students will use casting in different mathematical expressions. Students will learn to create programs using variables, objects, and methods. (VAR) Students will use a compiler to create a complete java program. Students will find and correct syntax errors and runtime errors. (VAR) Students will create and use Integer and Double objects. (VAR) Students will recognize the social and ethical implications of computer use. **LO** - VAR-1.A, VAR-1.B, VAR-1.C, VAR-1.F *Materials : Videos, Slides, Worksheets, etc.* *Labs : Write a pseudocode program for a given assignment. Use the pseudocode to create the program while also learning how to fix syntax and runtime errors.* (Skill 4.B) *Labs : Create a program that uses print and println to create ASCII shapes.* (Skill 1.B) *Labs : Create a program that reads in values, stores values, and prints out values. The program will use variables and operators to add, subtract, multiply, and divide. The program will use mod for remainders. The program must use Scanner to read in values. Students will use several different data types, including int, double, and String.* (Skill 2.A) (Skill 3.A) (VAR) *Activities : Pair programming, group code analysis* *Discussion : Computer Science article discussions and analysis. Students will find articles that show the positive and negative impacts of innovations related to computer science.* *Assessments : Labs, Quizzes, and Tests* **Big Ideas –** 2.Variables **Thinking Practices –** 1.Program Design and Algorithmic Development, 2.Code Logic, 3.Code Implementation, 4.Code Testing

2 weeks	**Using Objects, Methods, and Classes** **Course Topics [CED Unit Topics 2.1 - 2.5 and 2.9]** Classes Objects Constructors Methods and Parameters Void and Return (non-void) methods Math class methods **Student Objectives** Students will learn to create programs using variables and objects. Students will differentiate between a class and an object. (MOD) Students will create and call constructors and methods. (MOD) Students will instantiate objects and call methods. (MOD) Students will create objects by calling constructors. (MOD) Students will learn the differences between void and return methods. (MOD) Students will call methods with and without parameters. (MOD) Students will learn to use the Math class static return methods, including sqrt, abs, pow, and random. Students will learn to create programs using classes, methods, and parameters **LO -** MOD-1.B, MOD-1.C, MOD-1.D, MOD-1.E, MOD-1.F, MOD-1.G **LO -** MOD.1-H, CON-1.D *Materials : Videos, Slides, Worksheets, etc.* *Activites : Work individually or in pairs to evaluate code segments [worksheets] with simple method calls, multiple method calls, and methods calling other methods.* (Skill 2.C) *Activites : Students will use the Java docs to research the Math class and Math class methods. Students will need to find several useful Math methods beyond those in the AP subset. Students will learn to use the Java docs to better understand what provided Java methods exist and what exactly they can do.* (Skill 5.A) *Labs : Write programs that use common math formulas [using math operators] to practice calculating values. Some of the programs will use common Math class methods such as sqrt and pow. Some programs will calculate areas and perimeters of common shapes.* (Skill 1.B) (MOD) *Labs : Write programs that have classes with instance variables, constructors, and methods. Make a Dog class and a Triangle class tested with provided runner code. Students will be asked to create additional test cases beyond those provided.* (Skill 3.B) (MOD) *Labs : Interact with Jeroo by using objects and writing object code. Write programs to make Jeroo move around an island use the Jeroo methods.* (Skill 1.C) *Activities : Pair programming, group code analysis* *Assessments : Labs, Quizzes, and Tests* **Big Ideas –** 1.Modularity, 3.Control **Thinking Practices –** 1.Program Design and Algorithmic Development, 2.Code Logic, 3.Code Implementation, 5.Documentation

1 week	**Strings** **Course Topics [CED Unit Topics 2.6 and 2.7]** References Objects Methods Parameters Strings String Objects String methods **Student Objectives** Students will learn to create programs using objects and methods. Students will instantiate String objects and assign String references to values. (VAR) Students will use string literals. (VAR) Students will write programs that use strings, construct strings, set string values, return strings, and use string methods – indexOf(), substring(), length(), equals(), and compareTo(). Students will learn more about Strings objects and String methods. (VAR) **LO -** VAR.1-E, VAR.1-F *Materials : Videos, Slides, Worksheets, etc.* *Warmups : CodingBat [String 1]* *Labs : Write programs with methods that will return the first, last, and middle letters of a string. Write programs with methods that will return a string with the first and last letters switched as well as removing certain letters. Students will use the String class and call numerous String class methods.* (Skill 2.C) (Skill 3.A) (VAR) *Activites : Students will use the Java Docs to learn more about the String class. Students will find one new method to explain to a partner.* *Activities : Pair programming, group code analysis* *Assessments : Quizzes and Tests* **Big Ideas –** 2.Variables **Thinking Practices –** 2.Code Logic, 3.Code Implementation

2 weeks	**Boolean Logic and Conditionals** **Course Topics [CED Unit Topics 3.1 - 3.7]** Boolean Logic If Statements, If Else Statements, If Else If Statements, Nested ifs Logical and relational operators Boolean expressions Comparing Objects Algorithms **Student Objectives** Students will learn to create programs using classes, methods, and parameters. Students will learn to use boolean variables, learn to use boolean laws, and learn to solve logic problems with boolean expressions. Students will evaluate boolean expressions using relational operators. Students will solve problems using relational and logical operators. Students will represent branching logical processes. (CON) Students will use conditionals (if, if else, if else if) to solve problems. (CON) Students will design if statements, design if else if statements, create ifs to check values, and create ifs that check multiple conditions. (CON) Students will use nested if statements. (CON) Students will compare and contrast boolean expressions. Students will evaluate compound boolean expressions. Students will design and create algorithms using ifs and strings. (CON) Students will use if statements to compare references and the values of objects. (CON) Students will create algorithms to determine odd, even, and check numeric digits. (CON) **LO -** CON-1.E, CON-2.A, CON-2.B, CON-1.F, CON-1.G, CON-1.H *Materials : Videos, Slides, Worksheets, etc.* *Warmups : CodingBat [Logic 1 and Logic 2]* *Labs : Write a program with a method that checks to see if the first letter in a string appears multiple times. Write several other programs that use loops and if statements to search strings for specified values and / or create new strings using new and existing values.* (Skill 3.C) *Labs : Create a program using graphics that creates an Etch-A-Sketch. Events will be used to trap for key presses in order to draw lines to make a shape. If statements will be used to test conditions in order to move and draw the lines. Students will be writing new code that works with existing Java graphics and event handling code. Students will be provided with test cases. Students will determine other ways / cases to test the final program.* (Skill 1.C) (Skill 3.C)(Skill 4.A) (CON) *Activities : Pair programming, group code analysis* *Assessments : Quizzes and Tests* **Big Ideas –** 3.Control **Thinking Practices –** 1.Program Design and Algorithmic Development, 3.Code Implementation, 4.Code Testing

3 weeks	**Boolean Logic and Loops** **Course Topics [CED Unit Topics 4.1 – 4.5]** Loops – for and while and nested loops Logical and relational operators Analysis of algorithms and algorithm design Informal comparisons of running times Exact calculation of statement execution counts **Student Objectives** Students will learn to create programs using classes, methods, and parameters. Students will learn to use boolean variables, use boolean laws, and solve logic problems with boolean expressions. Students will solve problems using relational and logical operators. Students will use while and for loops to create algorithms to solve problems. (CON) Students will create loops that increase and decrease in value. (CON) Students will use nested loops. (CON) Students will design and create algorithms using that incorporate ifs and loops. (CON) Students will design and create algorithms using loops and strings. (CON) Students will create algorithms to sum values, count values, check numeric digits, determine odd / even, average values, and find min and max values. (CON) Students will analyze algorithms, compare run times, and calculate statement execution counts. **LO -** CON-1.E, CON-2.A, CON-2.B, CON-1.F, CON-1.G, CON-1.H **LO -** CON-2.C, CON-2.D, CON-2.E, CON-2.F, CON-2.G, CON-2.H *Materials : Videos, Slides, Worksheets, Magpie, Consumer, etc.* *Warmups : CodingBat [String 2]* *Activites : Work in pairs to evaluate code segments that use loops [worksheets] to determine the output and order of the statement execution.* (Skill 2.B) *Labs : Write a program that will use graphics and events to see which mouse button was pressed. The program will display different shapes for each of the mouse buttons. Test code will be provided and students will be expected to generate additional test cases. Students will need to work with existing code.* *Labs : Write a guessing game lab that use loops and random numbers to see if a user can guess a secret number. The program will keep stats on guesses and accuracy. Scanner will be used to read in use guesses.* (Skill 3.C)(CON) *Activities : Students will read the Magpie student manual and complete activities 1-4 or students can work through the Consumer lab activities. Students will analyze provided code and make adjustments to existing code. Work individually or in groups. Students will complete the open-ended Consumer Lab project.* (Skill 3.A) (CON) *Activities : Pair programming, group code analysis* *Assessments : Quizzes and Tests* **Big Ideas –** 3.Control **Thinking Practices –** 1.Program Design and Algorithmic Development, 2.Code Logic, 3.Code Implementation, 4.Code Testing

3 weeks	**Designing and Writing Classes** **Course Topics [CED Unit Topics 5.1 – 5.10]** Abstraction Classes and instance Variables Scope Constructors Accessor / Mutator Methods Static variables and methods This Object Oriented development Top-down development Social and Ethical Implications of Computer Use **Student Objectives** Students will learn to create programs using classes, methods, parameters, abstraction, and all aspects of object oriented programming. (MOD) Students will design classes. (MOD) Students will use public and private to designate access to class members. Students will declare instance variables, write constructors, and write methods. (MOD) Students will comment code to provide clarity. Students will design and create mutator and accessor methods with and without parameters. Students will design and create static methods. (MOD) Students will use static and this when accessing variables and methods. (MOD) Students will recognize the scope of variables and parameters. (VAR) Students will recognize the social and ethical implications of computer use. (IOC) **LO -** MOD-2.A, MOD-3.A, MOD-2.B, MOD-2.C, MOD-2.D, MOD-2.E, MOD-2.F **LO -** MOD-2.G, MOD-2.H, VAR-1.G, VAR-1.H, IOC-1.A *Materials : Videos, Slides, Worksheets, Celebrity Lab, etc.* *Labs : Design and create classes to simulate the behaviors of a Car and a CookieJar. Students will determine which variables and methods are needed. Some runner test code will be provided, but students will be expected to add additional test cases as well.* (Skill 3.B) (Skill 4.A) (MOD) *Labs : Write a Shape class that uses graphics to create a resizable shape. The Shape class will have instance variables, constructors, methods, and a toString.* (Skill 3.B) (MOD) *Labs : Students will read the Celebrity Lab manual and work activities. Students will complete the open-ended Celebrity lab project.* (Skill 3.B)(MOD) *Activities : Pair programming, group code analysis* *Activites / Labs : Students will work individually or in pairs to analyze and write code on paper for past Class Creation FRQS. Students will use provided code templates to type up and test the hand-written code. Runners with test cases are provided.* (Skill 1.B) (Skill 3.B) (Skill 5.D) *Discussion : Students will find articles that show the positive and negative impact of computer use and computer science on society and bring in for discussion. Data privacy will be a focus.* *Assessments : Quizzes and Tests* **Big Ideas –** 1.Modularity, 2.Variables, 4.Impact of Computing **Thinking Practices –** 3.Code Implementation, 4.Code Testing, 5.Documentation

2 weeks

Arrays and Array Processing

Course Topics [CED Unit Topics 6.1 – 6.4]

One-dimensional arrays
Array Creation
Traversals
Insertions
Deletions
Algorithms

Student Objectives

Students will declare 1D arrays, set array values, search for values in an array, and design algorithms that traverse and manipulate array values.   (VAR)

Students will use arrays to store primitive and reference types. (VAR)

Students will use different types of loops to traverse and process values in an array, including a traditional for loop and an enhanced for loop. (VAR)

Students will evaluate and implement commonly used algorithms: such as but not limited to, find biggest, find smallest, search, count occurrences, and remove specified values. (CON)

Students will create classes that contain arrays. (VAR)

LO - VAR-2.A, VAR-2.B, VAR-2.C, CON-2.1

Materials: Videos, Labs, Slides, Worksheets, etc.
Warmups : CodingBat [Array 1 and Array 2]
Labs : Write a program with a method that will determine the distance between the first odd number and last even number in an array. (Skill 2.A) (Skill 3.C) (Skill 3.D) (VAR)
Labs : Write a method that will determine if any number in the array repeats. The program will use ifs and loops to traverse and check the array values. (Skill 3.C) (Skill 3.D) (CON)
Labs : Create a class that contains an array instance variable that simulates the behavior of an elevator. The class can determine how many times a particular floor has been visited as well as several other operations. The class will contain an array instance variable, constructors, methods, and a toString. Students will work with provided runner code with test cases as well as creating their own thorough cases to test the code. (Skill 1.B) (Skill 3.C) (Skill 3.D) (Skill 4.A) (VAR)
Activites / Labs : Students will work individually or in pairs to analyze and write code on paper for past Array FRQS. Pre and post conditions will be discussed in greater detail. Students will use provided code templates to type up and test the hand-written code. Runners with test cases are provided. (Skill 1. B) (Skill 3.D) (Skill 5.D)
Activities : Pair programming and group code analysis sessions.
Assessments : Labs, Quizzes, and Tests

Big Ideas – 2.Variables, 3.Control

Thinking Practices – 1.Program Design and Algorithmic Development, 2.Code Logic, 3.Code Implementation, 4.Code Testing

2 weeks	**Lists and List Processing - ArrayList class** **Course Topics [CED Unit Topics 7.1 – 7.7]** Object Oriented Programming ArrayList class basics and methods ArrayList Traversals, Insertions – add, and Deletions – remove ArrayList algorithms – searching and sorting Choose appropriate data representation and algorithms Identify boundary cases and generate appropriate test data Social and Ethical Implications of Computer Use **Student Objectives** Students will instantiate ArrayList [list] objects. (VAR) Students will set list values, get list values, remove list values, search for values in a list, create algorithms that manipulate list values, and use different types of loops to process values in a list. (CON) Students will use the ArrayList methods – set, get, remove, size, and clear. (VAR) Students will traverse ArrayLists with standard for loops and enhanced for loops. (CON) Students will implement commonly used algorithms: such as, but not limited to find biggest, find smallest, sum, search, count occurrences, and remove specified values. Students will learn the linear search / sequential search algorithm. (CON) Students will learn the selection and insertion sort algorithms. (CON) Students use the 11s lab or equivalent lab to gain a better understanding of how to work with a list of classes. Students will recognize the social and ethical implications of computer use. (IOC) **LO -** VAR-2.D, VAR-2.E, CON-2.J, CON-2.K, CON-2.L, CON-2.M, IOC-1.B *Materials : Videos, Labs, Slides, Worksheets, Elevens lab, Data Lab, etc.* *Labs : Write arraylist programs that will get the 1st x values from an array, generate a list of prime numbers, and return a list of all factors of a number.* (Skill 1.B) *Labs : Write an arraylist class program that simulates the behavior of a flower garden that can return statistics of particular types and colors of flowers. The program will contains an ArrayList instance variable, constructors, methods, and a toString.* (Skill 3.D) (VAR) *Labs : AP CS A Lab - 11s lab units 1-4, Data Lab* *Activites / Labs : Students will work individually or in pairs to analyze and write code on paper for past ArrayList FRQS. Pre and post conditions will be discussed in more detail. Students will use provided code templates to type up and test the hand-written code. Runners with test cases are provided.* (Skill 1.B) (Skill 3.D) (Skill 5.D) *Activities : Students will read the Elevens student manual and complete activities 1-4 or students could work through the Data Lab. Work individually or in groups. Students will complete the open-ended Data Lab project.* (Skill 3.D) (VAR) (IOC) *Activities : Pair programming and group code analysis sessions.* *Discussion : Students will find articles for discussion related to data privacy and the risks / rewards of storing and protecting large sets of data and private information.* *Assessments : Labs, Quizzes, and Tests(m/c and free response)* **Big Ideas –** 2.Variables, 3.Control, 4.Impact of Computing **Thinking Practices –** 1.Program Design and Algorithmic Development, 2.Code Logic, 3.Code Implementation, 4.Code Testing
	END OF SEMESTER ONE

2 weeks	**Matrices [2D arrays] and Matrix Processing** **Course Topics [CED Unit Topics 8.1 and 8.2]** Arrays [1D and 2D arrays] Traversals Insertions Deletions 2D array algorithms Test classes and libraries in isolation Choose appropriate data representation and algorithms Identify boundary cases and generate appropriate test data **Student Objectives** Students will create matrices that store primitive and reference data. (VAR) Students will traverse matrices using standard and enhanced for loops. (CON) Students will use arrays [1d arrays] and arrays of arrays [2d arrays], learn how to find a values, print the values in an array of arrays [2d array], change the values in an array of arrays, and develop small and large algorithms that manipulate an array of arrays. (VAR) Students will implement commonly used algorithms: such as, but not limited to find biggest, find smallest, sum, search, count occurrences, and remove specified values. (CON) Students use the Picture lab or a comparable lab to gain a better understanding how to work with a large program, modify existing code from a large provided program, and how to expand existing code. **LO -** VAR-2.F, VAR-2.G, CON-2.N *Materials: Videos, Labs, Slides, Worksheets, PictureLab, Stenography, etc.* *Labs: Write programs that will sort a matrix, count a specified value, change values in a matrix.* *Labs : Write programs that will simulate the game of tic-tac-toe. The game will required complete class creation with matrix instance variables, constructors, methods, and toString methods. The game will check a provided board to see which player wins the game. Ifs and loops will be required for this program. Test cases will be provided.* (Skill 1.B) (Skill 3.E) (Skill 4.A) (VAR) (CON) *Labs : Write programs that will take a string and make a square pattern in the matrix with the letters from the String. The same process can be used to make different string shapes. Each program will have matrix instance variables, constructors, methods, and a toString. Students will be asked to complete all methods and constructors. Runner files with test code will be provided, but students will be asked to add / create additional test cases.* (Skill 1.B) (Skill 3.E) (Skill 4.A) (VAR) *Labs: Students will read the PictureLab & Stenography manuals and complete activities. Students will complete the open-ended Stenography project.* (Skill 3.E) (VAR) *Activites / Labs : Students will work individually or in pairs to analyze and write code on paper for past Matrix [2d array] FRQS. Students will use provided code templates to type up and test the hand-written code. Runners with test cases are provided.* (Skill 1. B) (Skill 3.E) *Activities : Pair programming and group code analysis sessions.* *Assessments : Labs, Quizzes, and Tests* **Big Ideas –** 2.Variables, 3.Control **Thinking Practices –** 1.Program Design and Algorithmic Development, 2.Code Logic, 3.Code Implementation, 4.Code Testing

3 weeks	**Extending Classes** **Course Topics [CED Unit Topics 9.1 - 9.7]** Abstraction Object Oriented Programming Multi-class programming Extending classes Super classes and sub classes Creating hierarchies Overriding methods Polymorphism Test classes and libraries in isolation **Student Objectives** Students will learn how to write programs that work with several classes. (MOD) Students will learn how to extend a class to make a new class. (MOD) Students will create sub and super classes. (MOD) Students will create a relationship between sub and super classes. Students will define super class variables that refer to sub class objects. Students will call methods and super and sub classes. (MOD) Students will learn about the Object super class and call Object methods. (MOD) Students will learn about polymorphism. Students use the 11s lab or comparable lab to gain a better understanding how to work with a large program, modify existing code from a large provided program, and how to expand existing code. **LO -** MOD-3.B, MOD-3.C, MOD-3.D, MOD-3.E *Materials : Videos, Labs, Slides, Worksheets, Elevens, Celebrity* *Labs : Students will design and create a graphical pong game that has classes to represent an underlying base Block object, a Ball object, and Paddle objects. The game can be expanded using inheritance to make a ball that speeds up and one that changes colors randomly. The game can be changed to make several other ball based games like brick-breaker. Runners with test cases will be provided and students will be required to generate test cases of their own to ensure each class works as intended before moving on to the next Class. Block must be fully tested before moving on to Ball and so on.* (Skill 1.B) (Skill 3.B) (Skill 4.A) (MOD) *Labs : Students will design and create a BlackJack game that has Card, Deck, Player, and Dealer classes. The final game requirement is to add a multi-player option. The game uses inheritance to extend the Card to make different types of cards for different card games. Students may choose to create Blackjack or another card game. An ArrayList<Player> will be used to create a multi-player game. Students must be able to explain why making a Card and extending it is a good approach or bad approach. Students will be required to create thorough test cases as well as work with provided test cases and existing code.* (Skill 1.B) (Skill 3.B) (Skill 5.C) (MOD) *Labs : Students will read the Elevens student manual and work through the inheritance activities.* *Labs : If not already completed in a prior unit, students will read the Celebrity Lab manual and work activities. Students will complete the open-ended Celebrity lab project.* (Skill 3.B)(MOD) *Activites : Analyze past year's free response questions* *Activities : Pair programming, group code analysis* *Assessments : Labs, Quizzes, and Tests* **Big Ideas –** 1.Modularity **Thinking Practices –** 1.Program Design and Algorithmic Development, 3.Code Implementation, 4. Code Testing, 5. Documentation

2 weeks	**Recursion and Searching and Sorting** **Course Topics [CED Unit Topics 10.1 and 10.2]** Recursion Recursive algorithms Searching - Sequential Search and Binary Search Sorting - Selection Sort, Insertion Sort, and Merge Sort Identify boundary cases and generate appropriate test data Choose appropriate data representation and algorithms Analysis of algorithms - iInformal comparisons of running times Exact calculation of statement execution counts. **Student Objectives** Students will learn to use recursion to solve problems, learn to use recursion to find values, and learn when and when not to use recursion. (CON) Students will write recursive code and analyze provided recursive code. (CON) Students will determine the answers generated by provided recursive methods. Students will learn about sorting and searching, learn all common sorts, learn all common sorts, learn to identify sorts and searches, learn to write all of the AP CS A sorts and searches, and learn when to use a particular search and sort for a specific situation. (CON) Students will use recursive sorting algorithms with strings, arrays, matrices, and ArrayList. **LO -** CON-2.O, CON-2.P *Materials: Videos, Labs, Slides, Worksheets, etc.* *Warmups : CodingBat [Recursion 1 and Recursion 2]* *Activites : Work in pairs to evaluate code segments that use recursion [worksheets] to determine the output and order of the statement execution.* (Skill 2.B) *Activity : Compare and contrast iterative and recursion solutions for linear and binary search to better understand how the algorithms accomplish the same tasks but with different approaches. Evaluate how iterative and recursion solutions are similar, but different.* (Skill 4.C) *Activity : Work through sorting algorithms in pairs. Examine the code and efficiency of the sorts. Given a small array to sort, determine the number of swaps and discuss the code used. Sort playing cards to demonstrate understanding of the sorting algorithms. Explain to a partner how and why the sorts manipulate / move the cards in the manner that they do.* (Skill 2.D) (Skill 5.A) (CON) *Labs : Create recursive methods to change and count values in a matrix.* (Skill 3.E) *Labs : Review several sorting algorithms and adapt them for use with matrices.* (Skill 5.C) *Labs : Create a program that uses recursion to solve a maze using recursive calls to move up, down, left, and right. There are options to extend the lab by adding algorithms to determine the shortest path and to use graphics to display the movement in the maze. This program will use include a matrix instance variable, constructors, methods, and a toString. Students will be asked to make additional test cases as well as work with existing code.* (Skill 1.B) (Skill 3.E) (CON) *Labs : Create a graphical program that will generate a fractal using recursion. Students will research common fractal patterns and select a fractal of their choice.* *Activities : Pair programming, group code analysis* *Assessments : Labs, Quizzes, and Tests* **Big Ideas –** 3.Control **Thinking Practices –** 1.Program Design and Algorithmic Development, 2.Code Logic, 3.Code Implementation, 4.Code Testing, 5.Documentation

11 weeks	**Review for the AP Computer Science Exam** *Core Review Topics* *Writing Classes* *Arrays - 1D and 2D arrays* *ArrayList – ArrayList< SomeClass >* *Inheritance and Classes* *AP CS A Labs – Magpie, Elevens, and Picture Lab* *AP CS A Labs – Consumer, Data Lab, Stenography, and Celebrity* *Readings : Past year's free response and multiple choice questions* *Readings : Review book units - Baron's AP CS A Review Book* *Activites / Labs : Students will work individually or in pairs to analyze and write code on paper for past FRQS. Students will use provided code templates to type up and test the hand-written code. Runners with test cases are provided.* *Videos : Use videos to review concepts as needed.* *Labs : AP CS A Labs – Magpie, Elevens, Picture Lab* *Labs : AP CS A Labs – Consumer, Data Lab, Stenography, and Celebrity* **Thinking Practices** Practice 1 - Program Design and Algorithm Development Determine required code segments to produce a given output. Practice 2 - Code Logic Determine the output, value, or result of given program code given initial values. Practice 3 - Code Implementation Write and implement program code. Practice 4 - Code Testing Analyze program code for correctness, equivalence, and errors. Practice 5 - Documentation Describe the behavior and conditions that produce identified results in a program **Big Ideas** Big Idea 1 - Modularity Big Idea 2 - Variables Big Idea 3 – Control Big Idea 4 – Impact of Computing
	End of Semester Two

AP Computer Science A: Curricular Requirements

CR1	Students and teachers have access to a college-level textbook in print or electronic format. **[Pages ALL]**
CR2	The course provides opportunities to develop student understanding of the required content outlined in each of the units described in the AP Course and Exam Description (CED). **[Pages ALL]**
CR3	The course provides opportunities to develop student understanding of the big ideas. **[Pages ALL]**
CR4	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 1 : Program Design and Algorithm Development. **[Pages 2,3,5,6,8,9,10,12]**
CR5	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 2 : Code Logic. **[Pages 2,3,4,6,8,9,10,12]**
CR6	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 3 : Code Implementation. **[Pages ALL]**
CR7	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 4 : Code Testing. **[Pages 2,5,6,7,8,9,10,11,12]**
CR8	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 5 : Documentation. **[Pages 3,7,11,12]**
CR9	This course provides students with hands-on lab experience to practice programming through designing and implementing computer-based solutions to problems. **[Pages ALL]**

AP Computer Science A: Resources Requirements

The school ensures that each student has a college-level text for individual use inside and outside of the classroom and has access to the AP Computer Science A Labs.

The school ensures that each student has access to a computer for at least three hours a week; three hours are the bare minimum, additional time is desirable. The computer system must contain appropriate software to create and edit programs and must allow programs comparable in size to the current AP Computer Science A Labs to compile in seconds. Internet access is strongly encouraged.

Classroom Management

AP Computer Science should be taught in a computer lab where each student has access to a computer each and every day.

Each student will spend at least 20 hours in class on the computer writing programs and working with programming assignments / programming labs.

Each student in each class must have access to a PC during class, before school, and after school.

Each topic covered should allow for multiple programming opportunities. The AP CS A Labs are quite helpful in this area. Codingbat is also a nice resource. The students should be allowed to work on programming assignments in groups or pairs when possible. Each student must be able to write programs, but group projects are beneficial.

Each student will spend at least 20 hours in class on the computer writing programs and working with programming assignments / programming labs.

The school ensures that each student has a college-level text for individual use inside and outside of the classroom and has access to the AP Computer Science A labs. The school ensures that each student has access to a computer for at least three hours a week; three hours are the bare minimum, additional time is desirable.

The computer system must contain appropriate software to create and edit programs and must allow programs comparable in size to the current AP Computer Science A labs to compile in seconds. Internet access is strongly encouraged.

Each student will spend at least 20 hours in class on the computer writing programs and working with programming assignments / programming labs.