AP Computer Science A introduces students to computer science through programming. Fundamental topics in this course include the design of solutions to problems, the use of data structures to organize large sets of data, the development and implementation of algorithms to process data and discover new information, the analysis of potential solutions, and the ethical and social implications of computing systems. The course emphasizes object-oriented programming and design using the Java programming language.

PLTW Computer Science A Release 2020-2021
PLTW's Computer Science A is a full-year course recommended for students in grades 11–12. The course aligns to the CSA Framework. Additionally, PLTW courses are designed to prepare students to thrive in college, careers, and beyond. As a result, many students choose to take AP exams to demonstrate the knowledge and skills they've gained to colleges and universities.

Exploring Careers in Computer Science
In Computer Science A, students develop the in-demand computer science skills critical to thrive in any of today's and tomorrow's careers. The course promotes computational thinking and Java coding fundamentals and introduces computational tools that foster creativity. It aims to build students' awareness of the tremendous demand for computer scientists and those who have computational thinking skills, and engages students to consider issues raised by the impact of computing on society. Each unit also focuses on one or more computer science-specific career paths.

The Big Ideas of Computer Science A
PLTW Computer Science A provides students opportunities to develop understanding of each of the five Big Ideas described in the CSA AP Course and Exam Description. While the CSA Big Ideas are spiraled throughout the course, each is highlighted in a specific project or problem.
The "Big Ideas" in Computer Science Principles Include:

- **(MOD) Modularity**
- **(VAR) Variables**
- **(CON) Control**
- **(IOC) Impact of Computing**

The Computational Thinking Practices of Computer Science A
PLTW Computer Science A provides students opportunities to apply computational thinking practices described in the CSA AP Course and Exam Description. While the CSA Computational Thinking Practices are spiraled throughout the course, each is highlighted in a specific project or problem. The "Computational Thinking Practices"" in Computer Science A include:

- **(CTP1) Program Design and Algorithm Development**
- **(CTP2) Code Logic**
- **(CTP3) Code Implementation**
- **(CTP4) Code Testing**
- **(CTP5) Documentation**

Chapter 1 Java Coding Fundamentals (44 days) (27%)

In Chapter 1, students are introduced to **(MOD) Modularity, (VAR) Variables, (CON) Control, (IOC) Impact of Computing, and the Computational Thinking Practices** of successful computer science professionals. Students learn the fundamentals of coding in Java.

Unit 1 Primitive Types (10 Days)

Variables are used to store data within a computer. One category of data that can be stored in a variable is primitive data. In this unit, students are introduced to three of the primitive data types defined in Java™. They learn how to create variables to store values of these different data types and the basic operations that can be performed on them. Students also learn how to output data using a basic form of output through the System.out object.

Activity 1.1.1	Why Programming Why Java?	(1 day)
Activity 1.1.2	Variables and Data Types	(1 day)
Activity 1.1.3	Expressions and Assignment Statements	(1 day)
Activity 1.1.4	Compound Assignment Operators	(1 day)
Activity 1.1.5	Casting and Ranges of Variables	(3 days)
Project 1.1.6	Numbers Riddle	(3 days)
	(VAR) Variables	
	(CTP3) Code Implementation	

Unit 2 Using Objects (17 days)

This unit introduces a new type of data: reference data. Reference data allows real-world objects to be represented in varying degrees specific to a programmer's purpose. This unit builds on students' ability to write expressions by introducing them to Math class methods to write expressions for generating random numbers and other more complex operations. In addition, strings and the existing methods within the String class are an important topic within this unit. Knowing how to declare variables or call methods on objects is necessary throughout the course.

Activity 1.2.1	Objects: Instances of Classes	(1 day)
Activity 1.2.2	Creating and Storing Objects	(2 days)
Activity 1.2.3	Calling a Void Method	(1 day)
Activity 1.2.4	Calling a Void Method with Parameters	(1 day)
Activity 1.2.5	Calling a Non-Void Method	(1 day)
Activity 1.2.6	String Objects: Concatenation, Literals, and More	(2 days)
Activity 1.2.7	String Methods	(2 days)
Activity 1.2.8	Wrapper Classes: Integer and Double	(2 days)
Activity 1.2.9	Using a Math Class	(2 days)
Project 1.2.10	Mad Libs®	(3 days)
	(MOD) Modularity	
	(CTP1) Program Design and Algorithm Development	

Unit 3 Boolean Expressions and if Statements (13 days)

Algorithms are composed of three building blocks: sequencing, selection, and iteration. This unit focuses on selection, which is represented in a program by using conditional statements. Conditional statements give the program the ability to decide and respond appropriately and are a critical aspect of any nontrivial computer

program. In addition to learning the syntax and proper use of conditional statements, students will build on the introduction of Boolean variables by writing Boolean expressions with relational and logical operators.

Activity 1.3.1	Boolean Expressions	(1 day)
Activity 1.3.2	If Statements and Control Flow	(1 day)
Activity 1.3.3	If/Else Statemnts	(1 day)
Activity 1.3.4	Else If Statements	(1 day)
Activity 1.3.5	Compound Boolean Expressions	(2 days)
Activity 1.3.6	Equivalent Boolean Expressions	(2 days)
Activity 1.3.7	Comparing Objects	(2 days)
Project 1.3.8	Choose Your Path **(CON) Control** **(CTP2) Code Logic**	(4 days)

Problem 1
Students will implement everything they have learned to design, plan, and collaboratively develop a solution that completes the functionality of a provided Escape Room style game.

Problem 1	Escape Room **(MOD) Modularity** **(CTP1) Program Design and Algorithm Development**	(4 days)

Chapter 2: Iteration and Classes (45 days) (28%)

In Chapter 2, builds on previous understanding of (CON) Control, (MOD) Modularity, (VAR) Variables as students create new, user-defined reference data types in the form of classes. Students will focus on identifying appropriate behaviors and attributes of real-world entities and organizing these into classes. The creation of computer programs can have extensive impacts on societies, economies, and cultures. The legal and ethical concerns that come with programs and the responsibilities of programmers are also addressed in this chapter.

Unit 4 Iteration
This unit focuses on iteration using while and for loops. This unit introduces several standard algorithms that use iteration. Knowledge of standard algorithms makes solving similar problems easier, as algorithms can be modified or combined to suit new situations. Iteration is used when traversing data structures such as arrays, ArrayLists, and 2D arrays.

Activity 2.4.1	While Loops	(2 days)
Activity 2.4.2	For Loops	(2 days)
Activity 2.4.3	Developing Algorithms Using	(2 days)
Activity 2.4.4	Strings Nested Iteration	(2 days)
Activity 2.4.5	Informal Code Analysis	(2 days)
Project 2.4.6	Consumer Review Lab **(CON) Control** **(CTP3) Code Implementation**	(8 days)

Unit 5 Writing Classes
This unit will pull together information from all previous units to create new, user-defined reference data types in the form of classes. The ability to accurately model real-world entities in a computer program is a large part of what makes computer science so powerful. This unit focuses on identifying appropriate behaviors and attributes of real-world entities and organizing these into classes. The creation of computer programs can have extensive impacts on societies, economies, and cultures. The legal and ethical concerns that come with programs and the responsibilities of programmers are also addressed in this unit.

Activity 2.5.1	Anatomy of a Class	(1 day)
Activity 2.5.2	Constructors	(2 days)
Activity 2.5.3	Documentation with Comments	(2 days)
Activity 2.5.4	Accessor Methods	(2 days)
Activity 2.5.5	Mutator Methods	(2 days)
Activity 2.5.6	This Keyword	(1 day)
Activity 2.5.7	Writing Methods	(2 days)
Activity 2.5.8	Static Variables and Methods	(2 days)
Activity 2.5.9	Scope and Access	(2 days)
Activity 2.5.10	Ethical and social Implications of Computing Systems	(2 days)
Project 2.5.11	Game of Nim **(MOD) Modularity** **(CTP5) Documentation**	(4 days)

Problem 2
At the end of this chapter students will implement everything they have learned to design, plan, and collaboratively develop an ad system for a social media sight.

Problem 2	Sell My Pet Food **(CTP2) Code Logic** **(IOC) Impact of Computing**	(4 days)

Chapter 3 Arrays (41 days) (25%)

In Chapter 2, students are dig deeper into to (VAR) Variables and (CON) Control while utilizing arrays to represent collections of data using a single variable. They will apply standard algorithms to arrays; however, these same algorithms are used with ArrayLists and 2D arrays as well. Students will also utilize standard searching and sorting algorithms. In this chapter, students will also learn about privacy concerns related to storing large amounts of personal data and about what can happen if such information is compromised.

Unit 6 Arrays
This unit focuses on data structures, which are used to represent collections of related data using a single variable rather than multiple variables. Using a data structure along with iterative statements with appropriate bounds will allow for similar treatment to be applied more easily to all values in the collection. Just as there are useful standard algorithms when dealing with primitive data, there are standard algorithms to use with data structures. In this unit, students apply standard algorithms to arrays; however, these same algorithms are used with ArrayLists and 2D arrays as well.

Activity 3.6.1	Array Creation and Access	(2 days)
Activity 3.6.2	Traversing Arrays	(2 days)
Activity 3.6.3	Enhanced Loops for Arrays	(1 day)
Activity 3.6.4	Developing Algorithms Using Arrays	(3 days)
Project 3.6.5	Memory Game	(3 days)
	(VAR) Variables	
	(CTP4) Code Testing	

Unit 7 ArrayList

Building on what students learned in Unit 6, data structures are helpful when storing multiple related data values. Arrays have a static size, which causes limitations related to the number of elements stored, and it can be challenging to reorder elements stored in arrays. The ArrayList object has a dynamic size, and the class contains methods for insertion and deletion of elements, making reordering and shifting items easier. Deciding which data structure to select becomes increasingly important as the size of the data set grows, such as when using a large real-world data set. In this unit, students will also learn about privacy concerns related to storing large amounts of personal data and about what can happen if such information is compromised.

Activity 3.7.1	Introduction to ArrayList	(1 day)
Activity 3.7.2	ArrayList Methods	(2 days)
Activity 3.7.3	Traversing ArrayLists	(1 day)
Activity 3.7.4	Developing Algorithms Using ArrayLists	(2 days)
Activity 3.7.5	Searching	(4 days)
Activity 3.7.6	Sorting	(4 days)
Project 3.7.7	Data Lab	(3 days)
	(IOC) Impact of Computing	
	(CTP3) Code Implementation	

Unit 8 2D Arrays

Previous concepts will be implemented with two-dimensional (2D) arrays in this unit. A 2D array is most suitable to represent a table. Each table element is accessed using the variable name and row and column indices. Unlike 1D arrays, 2D arrays require nested iterative statements to traverse and access all elements. The easiest way to accomplished this is in row-major order, but it is important to cover additional traversal patterns, such as back and forth or column-major.

Activity 3.8.1	2D Arrays	(3 days)
Activity 3.8.2	Traversing 2D Arrays	(5 days)
Project 3.8.3	Steganography Lab	(4 days)
	(CON) Control	
	(CTP4) Code Testing	

Problem 3
At the end of chapter problem, you'll pursue a question of interest to you. To accomplish this, you will pose a question, identify a data set that will help you answer the question, and develop a program to use the data set to gain information to help you answer your question. Be thinking about what question you would like to answer and whether there is data that could help you.

Problem 3	Data Lab Activity 4	(4 days)
	(CON) Control	
	(CTP3) Code Implementation	

Chapter 4 Inheritance and Recursion (33 Days) (20%)

In Chapter 4, students focus on (MOD) Modularity, and (CON) Control as students to categorize classes into hierarchies through inheritance. They will learn about the power of recursion, solving smaller or simpler versions of the same problem rather than attempting an iterative solution.

Unit 9 Inheritance
Creating objects, calling methods on the objects created, and being able to define a new data type by creating a class are essential understandings before moving into this unit. One of the strongest advantages of Java is the ability to categorize classes into hierarchies through inheritance. Certain existing classes can be extended to include new behaviors and attributes without altering existing code. These newly created classes are called subclasses. In this unit, students will learn how to recognize common attributes and behaviors that can be used in a superclass and will then create a hierarchy by writing subclasses to extend a superclass. Recognizing and utilizing existing hierarchies will help students create more readable and maintainable programs.

Activity 4.9.1	Creating Superclasses and Subclasses	(1 days
Activity 4.9.2	Writing Constructors for Subclasses	(3 days)
Activity 4.9.3	Overriding Methods	(2 days)
Activity 4.9.4	Super Keyword	(2 days)
Activity 4.9.5	Creating References Using Inheritance Hierarchies	(2 days)
Activity 4.9.6	Polymorphism	(2 day)
Activity 4.9.7	Object Superclass	(1 day)
Activity 4.9.8	Celebrity Lab	(4 days)
	(MOD) Modularity	
	(CTP1) Program Design and Algorithm Development	

Unit 10 Recursion

Sometimes a problem can be solved by solving smaller or simpler versions of the same problem rather than attempting an iterative solution. This is called recursion, and it is a powerful math and computer science idea. In this unit, students will revisit how control is passed when methods are called, which is necessary knowledge when working with recursion. In this unit, students will learn how to write simple recursive methods and determine the purpose or output of a recursive method by tracing.

Activity 4.10.1	Recursion	(2 days)
Activity 4.10.2	Recursive Searching and Sorting	(2 days)
Activity 4.10.3	Presenting Your Development Process	(3 days)
	(CON) Control	
	(CTP5) Documentation	

Problem 4

This Problem brings together all of the skills you have learned throughout the course. You will use the software development cycle you have used in other projects. In this activity, you will design and develop a program of your choosing.

Problem 4	Almost Anything!	(5 days)
	(MOD) Modularity	
	(CTP1) Program Design and Algorithm Development	