



PLTW Computer Science

Computer Science A | Course Outline

The Computer Science A course includes over 160 hours of hands-on, structured lab experiences to engage students in individual or group problem solving. Thus, it includes a substantial lab component in which students design solutions to problems, express their solutions precisely (e.g., in the Java programming language), test their solutions, identify and correct errors, and compare possible solutions.

Computer Science A (CSA) is designed to be a full-year (160-day) course implemented in the 12th grade. Students cultivate their understanding of coding through analyzing, writing, and testing code as they explore concepts like modularity, variables, and control structures. CSA is designed with alignment to the College Board Computer Science A framework. The course is also designed with strong connections to the Computer Science K12 Frameworks (CS K12), ISTE Standards, and the Computer Science Teachers Association K-12 Computer Science (CSTA K-12 CS) Level 3A Standards.

Fundamental topics in this course include the design and development of solutions that use control-structures, data structures, and object-oriented programming using the Java programming language, the analysis of potential solutions, and the ethical and social implications of computing systems.

Chapter 1 Java Coding Fundamentals (28%)

Chapter 2 Iteration and Classes (28%)

Chapter 3 Arrays (26%)

Chapter 4 Inheritance and Recursion (18%)

Chapter 1 Java Coding Fundamentals (45 days)

Chapter 1 welcomes students to the world of Java programming fundamentals. Students work with an embedded code editor and Microsoft Visual Studio Code to learn the fundamentals of coding in Java. At the end of the chapter 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.

Java Coding Fundamentals Unit Summary

Unit 1 Primitive Types (10 days)

Unit 2 Using Objects (17 days)

Unit 3 Boolean Expressions and if Statements (14 days)

Problem 1 (4 days)

Please note: The information included in this document is subject to change. As with all course materials, we will continue to update as more information becomes available.

Unit 1 Primitive Types

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)

Unit 2 Using Objects

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	PLTW Project - Mad Libs	(3 days)

Unit 3 Boolean Expressions and if Statements

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 Statements	(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	PLTW Project - Choose Your Path	(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	(4 days)
-----------	-------------	----------

Chapter 2: Iteration and Classes (44 days)

Unit 2 expands what students know to include iteration and writing classes.

Iteration and Classes Unit Summary

Unit 4 Iteration	(18 days)
Unit 5 Writing Classes	(22 days)
Problem 2	(4 days)

Unit 4 Iteration

This unit focuses on iteration using while and for loops. It 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 Strings	(2 days)
Activity 2.4.4	Nested Iterations	(2 days)
Activity 2.4.5	Informal Code Analysis	(2 days)
Project 2.4.6	Consumer Review Lab	(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 realworld 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 Commands	(2 days)
Activity 2.5.4	Accessor Methods	(2 days)
Activity 2.5.5	Mutator Methods	(2 days)
Activity 2.5.6	Writing Methods	(1 days)
Activity 2.5.7	Static Variables and Methods	(2 days)
Activity 2.5.8	Scope and Access	(2 days)
Activity 2.5.9	this Keyword	(2 days)
Activity 2.5.10	Ethical and social Implications of Computing	(2 days)
Project 2.5.11	PLTW Project - Game of Nim	(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 to help people sell their pet food.

Problem	Sell My Pet Food	(4 days)
---------	------------------	----------

Chapter 3 Arrays (44 days)

The goal of chapter 3 is to introduce students to storing and manipulating data using data structures, such as Arrays, ArrayLists, and 2D Arrays.

Arrays Unit Summary		
Unit 6	Arrays	(11 days)
Unit 7	ArrayList	(17 days)
Unit 8	2D Arrays	(12 days)
Problem 3		(4 days)

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 days)
Activity 3.6.4	Developing Algorithms Using Arrays	(3 days)
Project 3.6.5	PLTW Project - Memory Game	(3 days)

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 days)
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)

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 accomplish 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	Stenography Lab	(4 days)

Problem 3

At the end of chapter problem, students pursue a question of interest to them. To accomplish this, they will pose a question, identify a data set that will help them answer the question, and develop a program to use the data set to gain information to help them answer their question.

Data Lab Activity 4	(4 days)
---------------------	----------

Chapter 4 Inheritance and Recursion (29 days)

The goal of chapter 4 is to further student knowledge of advanced object-oriented concepts, including inheritance and recursion.

Inheritance and Recursion Unit Summary		
Unit 9	Inheritance	(17 days)
Unit 10	Recursion	(7 days)
Problem 4		(5 days)



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 days)
Activity 4.9.7	Object Superclass	(1 days)
Project 4.9.8	Celebrity Lab	(4 days)

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	PLTW Project	(3 days)

Problem 4

This Problem brings together all of the skills students have learned throughout the course. Students will use the software development cycle they have used in other projects. In this problem, they will design and develop a program of their choosing.

Almost Anything! (5 days)

Trademark Attribution: PLTW, Project Lead The Way, and the PLTW logo are registered trademarks of Project Lead The Way, Inc. Python and the Python logos are trademarks or registered trademarks of the Python Software Foundation. All other brand names, product names, or trademarks belong to their respective holders.

Please note: The information included in this document is subject to change. As with all course materials, we will continue to update as more information becomes available.