Mathematics Department

AP Computer Science A

AP Computer Science A is an introductory college-level computer science course. Students cultivate their understanding of coding through analyzing, writing, and testing code as they explore concepts like modularity, variables, and control structures. Students learn to design, write, and test Java programs to solve problems. (College Board, 2025)

Developed By: Monika Truty, College Board
Effective Date: September 2025

Scope and Sequence

September	**Summer Assignment: Basics of Java (First part of Unit 1)** -Introduction to Algorithms, Programming, and Compilers -Variables, Data Types -Expressions and Output -Assignment Statements and Input -Casting and Range of Variables -Compound Assignment Operators -Application Program Interface (API) and Libraries -Documentation with Comments **COMMON ASSESSMENT: Prerequisite TEST** **Unit 1: Using Objects and Methods** -Method Signatures -Calling Class Methods -Math Class -Objects: Instances of Classes -Object Creation and Storage (Instantiation)
October	**Using Objects and Methods (cont.)** -Calling Instance Methods -String Manipulation **Unit 2: Selection and Iteration** -Algorithms with Selection and Repetition -Boolean Expressions and comparing them -If Statements -Nested if Statements -Compound Boolean Expressions

November	**Unit 2: Selection and Iteration (cont.)** -While Loops -For Loops -Implementing Selection and Iteration Algorithms -Implementing String Algorithms -Nested Iteration -Informal Run-Time Analysis
December	**Unit 3: Class Creation** -Abstraction and Program Design -Impact of Program Design -Anatomy of a Class -Constructors -Writing Methods and passing/returning references of an object
January	**Class Creation (cont.)** -Class Variables and Methods -Scope and Access -this Keyword
February	**Unit 4: Data Collection** -Ethical and Social Issues Around Data Collection -Introduction to Using Data Sets -Array Creation and Access -Array Traversals -Implementing Array Algorithms -Using Text Files
March	**Data Collection (cont.)** -Wrapper Classes -ArrayList Methods -ArrayList Traversals -Implementing ArrayList Algorithms -2D Array Creation and Access -2D Array Traversals -Implementing 2D Array Algorithms -Searching Algorithms -Sorting Algorithms -Recursion -Recursive Searching and Sorting
April	**AP EXAM REVIEW** -Using AP Classroom, albert.io, practice tests
May	**AP EXAM REVIEW (cont.)** **AP EXAM** **FINAL ASSESSMENT Review**
June	**FINAL ASSESSMENT**

Unit 1

Using Objects and Methods

Summary and Rationale

Students will learn the core principles of object-oriented programming, beginning with foundational concepts like algorithms, programming, and compilers. Students learn to work with variables, data types, expressions, and inputs/outputs. Topics like casting, variable ranges, and the use of APIs and libraries are covered to promote efficient and modular programming. Coding practices like reusable code, proper documentation, method signatures, class methods and object instantiation are promoted to ensure efficient programming. Practical applications, including string manipulation and use of the Math class will help students solve real world problems.

Recommended Pacing

For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)	
AP - Algorithms & Programming	
3B-AP-10	Use and adapt classic algorithms to solve computational problems.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-12	Compare and contrast fundamental data structures and their uses.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules and/or objects.
3B-AP-16	Demonstrate code reuse by creating programming solutions using libraries and APIs.
3B-AP-20	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.
3B-AP-22	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
3B-AP-24	Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.
AP Computer Science A Conceptual Framework	
DAT-1	The way that the computer represents data is different from the way that the data are interpreted and displayed for the user. Programs are used to translate data into a representation that is more easily

	understood by people.
MOD-1	Objects are created from classes, which define the structure and behavior of a program. Understanding constructors, as well as the difference between static and instance methods, allows programs to use and interact with objects effectively.
MOD-2	APIs provide access to prewritten classes and methods that simplify programming tasks. By reading documentation and using libraries like Math and String, programmers can efficiently solve problems without writing all code from scratch.
VAR-1	Variables store data that can change as a program runs. Understanding how to declare, assign, and reference both primitive and object types is essential for managing data within a program.
OUT-1	Programs use output statements to communicate results and information to the user. Combining strings and variables in output helps make the displayed information clear and meaningful.
CSA Topics	
1.1	Primitive Data Types
1.2	Using Objects
1.3	Creating and Storing Objects
1.4	Using String Objects
1.5	Using APIs
1.6	Output
Interdisciplinary Connections	
Standard	
8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.AP.2	Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.5	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

Enduring Understandings:	**Essential Questions:**
Objects are instances of classes that encapsulate data and behavior; understanding how to create and use objects is fundamental to object-oriented programming. Primitive and reference data types store and represent information differently in memory, and proper use of	How can you represent buttons on a remote control using variables? How can you simulate election results using existing program code?

variables is essential for managing program state. Methods define the behavior of objects, with constructors used to initialize new objects; distinguishing between instance and static methods is key to proper method invocation. Using pre-existing classes and methods through APIs promotes modularity, efficiency, and code reuse, allowing programmers to build on existing functionality. Programs translate internal data representations into human-readable output, demonstrating the importance of data abstraction and effective communication between programs and users.	How can you use random numbers to add variety, excitement, and unpredictability to games?
Evidence of Learning (Assessments)	
Tests Quizzes Homework Programs Projects / Labs Class participation	
Objectives (SLO)	
Students will know: ● The difference between primitive and reference data types in Java. ● How to declare and assign variables, and understand how data is stored in memory. ● The structure and purpose of a class and how objects are created from classes. ● The role of constructors in initializing object data. ● How to call both static methods and instance methods appropriately. ● How to use String methods to manipulate and analyze text data. ● The value of using APIs and libraries (like Math and String) to simplify programming tasks. ● How to use output statements to display results and messages to the user. ● The importance of code documentation and using comments to explain program logic.	Students will be able to: ● Instantiate and use objects from a given class ● Call and write methods with and without parameters ● Use inheritance to create subclasses and extend functionality ● Implement algorithms using sequencing, selection, and iteration ● Debug programs using variables, outputs, and the problem-solving process ● Document code using comments, preconditions, and postconditions ● Collaborate with peers by giving and receiving feedback during code reviews ● Write readable, well-structured Java programs using classes, constructors, and methods
Suggested Resources/Technology Tools	

-Programming compiler codeHS
-Online notebook/journal
-AP CSA Practice Tests and Released Items
-AP Classroom
-Online Textbook/Practice Workbooks

Modifications

Special Education-
Modifications and accommodations must be aligned to students' IEPs and uphold expectations of the plan lawfully.
● Allow additional time for skill mastery.
● Provide small group/ individualized instruction.
● Allow use of assistive technology.
● Anticipate needs; pre-teach if needed.
● Have student repeat directions to check for understanding.
● Highlight text/ visual presentation.
● Modify assessment/assignment content, format, length (benchmarks, tests and quizzes) .
ML-
● Provide additional wait time for student responses to questions
● Provide simplification of sentence structure and repetition of questions/sentences exactly as stated before trying to rephrase.
● Speak clearly, using a normal tone and rate of speed, or slightly slower.
● Use directed reading activities. Ensure preview of text before assigned/read, provide pre-reading
Gifted and Talented-
● Advance, accelerate, or compact content
● Support abstract and advanced higher-level thinking through differentiated questioning
● Allowance for adapting of content for individual student interests
● Provide assignments geared to development in areas of affect, creativity, cognition, and research skills
504-
Modifications and accommodations must be aligned to the 504 plan and uphold expectations of the plan lawfully.
● Provide additional time for in-class assignments and assessments
● Allow for computer with voice output, spelling and grammar checker
● Offer individual help throughout the lesson
● Modify readings and primary source documents
At Risk-
● Provide small group instruction for identified areas for growth
● Allow for alternate pathways for completion of assignments.
● Provide additional wait time for student responses to questions
● Pair oral instruction with visuals
● Utilize manipulatives
For a full list of Accommodations and Modifications, please click here.

21ST CENTURY LIFE AND CAREER STANDARDS

- ❏ Apply appropriate academic and technical skills.
- ❏ Communicate clearly and effectively and with reason.
- ❏ Demonstrate creativity and innovation.
- ❏ Employ valid and reliable research strategies.
- ❏ Utilize critical thinking to make sense of problems and persevere in solving them.
- ❏ Model integrity, ethical leadership, and effective management.
- ❏ Plan education and career paths aligned to personal goals.
- ❏ Use technology to enhance productivity.

LINKS TO CAREERS: https://career.ku.edu/computer-science

Unit 2

Selection and Iteration

Summary and Rationale

Algorithms are composed of three building blocks: sequencing, selection, and iteration. While Unit 1 introduced sequencing, this unit focuses on selection and iteration. Selection is important to a computer program because it gives the programmer the ability to make a decision and respond to that decision using conditional statements. These allow programmers to incorporate choice into their programs: to create games that react to user interactions, to develop simulations that are more real world by allowing for variability, or to discover new knowledge in a sea of information by filtering out irrelevant data. Iteration is a form of repetition and changes the flow of control by repeating a segment of code. It is represented by while and for loops. In addition, students will build on the introduction of Boolean variables in Unit 1 by writing Boolean expressions with relational and logical operators. This unit also 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.

Recommended Pacing

For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)	
AP - Algorithms & Programming	
1B-AP-10	Develop programs using sequences, events, loops, and conditionals
2-AP-1	Analyze algorithms using loops and conditionals
2-AP-2	Develop and test programs with loops and conditionals
3A-AP-14	Create reusable code with parameters and control structures
3B-AP-15	Compare and refine algorithms involving iteration and selection
AP Computer Science A Conceptual Framework	
CON-1	Boolean expressions evaluate to true or false and are used to make decisions in a program.
CON-2	Conditional statements enable programs to choose different paths of execution based on conditions.
CON-3	Iteration allows a block of code to repeat as long as a condition is true.
CON-4	Effective algorithms often combine selection and iteration to solve complex problems efficiently.

CSA Topics	
2.1	Boolean Expressions
2.2	If Statements
2.3	If-Else Statements
2.4	Else-If Statements
2.5	Compound Boolean Expressions
2.6	Equivalent Boolean Expressions
2.7	Nested if Statements
2.8	While Loops
2.9	For Loops
2.10	Developing Algorithms Using Strings
2.11	Nested Iteration
2.12	Informal Code Analysis
Interdisciplinary Connections	
Standard	
NJSLSA.R7	Integrate and evaluate content presented in diverse media and formats.
HS-ETS1-2	Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems.
HSN.Q.A.1	Use units as a way to understand problems and guide the solution of multi-step problems.

Enduring Understandings:	**Essential Questions:**
Programs use Boolean expressions to make decisions about which parts of code to execute. Understanding how to evaluate and combine conditions is essential for controlling the logic of a program. Conditional statements like if and if-else allow a program to choose between different actions based on user input or program state. Loops (while, for) allow the program to repeat instructions until a condition is met, making programs more efficient and powerful.	Why is selection a necessary part of programming languages? How can you use different conditional statements to write a pick-your-own-path interactive story? How does iteration improve programs and reduce the amount of program code necessary to complete a task? Why are standard algorithms useful when solving new problems?

Algorithms that solve problems can be developed using a combination of decisions (selection) and repetition (iteration) to handle complex logic.	

Evidence of Learning (Assessments)

Tests
Quizzes
Homework
Programs
Projects / Labs
Class participation

Objectives (SLO)

Students will know:	Students will be able to:
● How to construct and evaluate Boolean expressions using relational (==, !=, <, >) and logical (&&, ||, !) operators. ● How to use if and if-else statements to control the flow of a program based on conditions. ● How to write and understand while and for loops for repeating actions in code. ● How to implement standard iterative algorithms like counting, summing, or finding minimum/maximum values. ● How to use loops to perform string-processing tasks such as searching, counting characters, or reversing strings. ● How nested loops work and how to use them effectively. ● How to perform informal runtime analysis by counting statement executions and comparing loop efficiencies. ● The value of using APIs and libraries (like Math and String) to simplify programming tasks. ● How to use output statements to display results and messages to the user. ● The importance of code documentation and using comments to explain program logic.	● Develop and implement algorithms using sequencing, selection, and iteration ● Write Boolean expressions and apply logical operators to control flow ● Use if, if-else, nested conditionals, and while/for loops to solve problems ● Trace and debug code that uses nested loops and complex conditionals ● Compare algorithm efficiency using informal runtime analysis ● Extend existing classes to create specialized subclasses ● Plan and write algorithms using pseudocode and decomposition ● Collaborate with peers to test and review code ● Write readable, well-structured Java programs using classes, constructors, and methods

Suggested Resources/Technology Tools

-Programming compiler codeHS
-Online notebook/journal
-AP CSA Practice Tests and Released Items
-AP Classroom

-Online Textbook/Practice Workbooks

Modifications

Special Education-
Modifications and accommodations must be aligned to students' IEPs and uphold expectations of the plan lawfully.
● Allow additional time for skill mastery.
● Provide small group/ individualized instruction.
● Allow use of assistive technology.
● Anticipate needs; pre-teach if needed.
● Have student repeat directions to check for understanding.
● Highlight text/ visual presentation.
● Modify assessment/assignment content, format, length (benchmarks, tests and quizzes) .
ML-
● Provide additional wait time for student responses to questions
● Provide simplification of sentence structure and repetition of questions/sentences exactly as stated before trying to rephrase.
● Speak clearly, using a normal tone and rate of speed, or slightly slower.
● Use directed reading activities. Ensure preview of text before assigned/read, provide pre-reading
Gifted and Talented-
● Advance, accelerate, or compact content
● Support abstract and advanced higher-level thinking through differentiated questioning
● Allowance for adapting of content for individual student interests
● Provide assignments geared to development in areas of affect, creativity, cognition, and research skills
504-
Modifications and accommodations must be aligned to the 504 plan and uphold expectations of the plan lawfully.
● Provide additional time for in-class assignments and assessments
● Allow for computer with voice output, spelling and grammar checker
● Offer individual help throughout the lesson
● Modify readings and primary source documents
At Risk-
● Provide small group instruction for identified areas for growth
● Allow for alternate pathways for completion of assignments.
● Provide additional wait time for student responses to questions
● Pair oral instruction with visuals
● Utilize manipulatives
For a full list of Accommodations and Modifications, please click here.

21ST CENTURY LIFE AND CAREER STANDARDS

- ❏ Apply appropriate academic and technical skills.
- ❏ Communicate clearly and effectively and with reason.
- ❏ Demonstrate creativity and innovation.
- ❏ Employ valid and reliable research strategies.
- ❏ Utilize critical thinking to make sense of problems and persevere in solving them.
- ❏ Model integrity, ethical leadership, and effective management.
- ❏ Plan education and career paths aligned to personal goals.
- ❏ Use technology to enhance productivity.

LINKS TO CAREERS: https://career.ku.edu/computer-science

Unit 3

Class Creation

Summary and Rationale

This unit will pull together information from the previous two 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. By being able to design their own classes, programmers are not limited to the existing classes provided within the Java libraries and can therefore represent their own ideas through classes. This unit focuses on identifying appropriate behaviors and attributes of real-world entities and organizing these into classes and their corresponding methods. The creation of computer programs can have an extensive impact 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.

Recommended Pacing

For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)	
AP - Algorithms & Programming	
2-AP-1	Design modular programs using classes and objects
3A-AP-16	Use objects and classes to model real-world scenarios
3A-AP-17	Understand constructors, methods, and instance variables in classes
3B-AP-18	Use abstraction to manage program complexity
IC - Impacts of Computing	
3B-IC-21	Understand ethical and social impacts of computing
AP Computer Science A Conceptual Framework	
MOD-1.A-F	Programs are developed by writing and combining procedures (methods and classes) that model behaviors.
VAR-1.A-B	Variables provide storage locations with associated types and names to hold data.
CRG-1.A-B	Classes can relate to one another through associations.
IOC-1.A-B	Methods enable communication with objects, including sending and receiving data.
CSA Topics	
3.1	Classes and Objects
3.2	Defining Classes with Instance Variables

3.3	Constructors
3.4	Instance Methods
3.5	this Reference
3.6	Overloading Methods
3.7	Access Modifiers (Public/Private)
3.8	Class Relationships
3.9	Designing Classes to Model Real-World Entities
3.10	Ethical and Social Implications of Software Development

Interdisciplinary Connections	
Standard	
ELA.RST.9-10.3	Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks, attending to special cases or exceptions defined in the text.
HS-ETS1-4	Use a computer simulation to model the impact of proposed solutions to a complex real-world problem.
NJSLS-MATH.MP.7	Use patterns and structure to help understand and connect mathematical concepts.

Enduring Understandings:	**Essential Questions:**
Classes and objects allow programmers to model real-world entities by encapsulating data (attributes) and behaviors (methods) into reusable blueprints. Understanding how to define classes, create objects, and organize data with instance variables is essential for building complex programs. Constructors initialize objects, setting their initial state, while methods provide functionality that operates on that state, enabling objects to perform actions and communicate. Class relationships, through object references, allow programs to represent interactions and build modular, maintainable code structures. Programmers must also consider ethical responsibilities, including how their software affects users and society, ensuring privacy, security, and fairness.	How can using a model of traffic patterns improve travel time? How can programs be written to protect your bank account balance from being inadvertently changed? How can you be a responsible programmer?

Evidence of Learning (Assessments)

Tests
Quizzes
Homework
Programs
Projects / Labs
Class participation

Objectives (SLO)

Students will know:

- How to design and document classes using UML.
- The use of instance variables and encapsulation with access modifiers.
- How constructors and the *this* keyword initialize and manage objects.
- Writing getter and setter methods to control object state.
- Differences between instance and local variables, and static members.
- How passing primitives and object references affects methods.
- Basic debugging strategies for improving code reliability.
- Ethical and societal impacts of programming, including privacy and security.

Students will be able to:

- Design and document Java classes using UML diagrams
- Declare instance variables and apply encapsulation using access modifiers
- Write and call constructors (with and without parameters)
- Use the this keyword to refer to instance variables and reduce ambiguity
- Write accessor (getter) and mutator (setter) methods to manage object state
- Identify the scope of variables within a class or method
- Use and explain the purpose of static variables and static methods
- Differentiate between passing primitives and object references as parameters
- Recognize and address system bugs to improve software reliability
- Evaluate the ethical and societal impacts of programs, including privacy, security, and AI

Suggested Resources/Technology Tools

-Programming compiler codeHS
-Online notebook/journal
-AP CSA Practice Tests and Released Items
-AP Classroom
-Online Textbook/Practice Workbooks

Modifications

Special Education-
Modifications and accommodations must be aligned to students' IEPs and uphold expectations of the plan lawfully.

- Allow additional time for skill mastery.
- Provide small group/ individualized instruction.
- Allow use of assistive technology.
- Anticipate needs; pre-teach if needed.
- Have student repeat directions to check for understanding.
- Highlight text/ visual presentation.

● Modify assessment/assignment content, format, length (benchmarks, tests and quizzes) .

ML-

● Provide additional wait time for student responses to questions
● Provide simplification of sentence structure and repetition of questions/sentences exactly as stated before trying to rephrase.
● Speak clearly, using a normal tone and rate of speed, or slightly slower.
● Use directed reading activities. Ensure preview of text before assigned/read, provide pre-reading

Gifted and Talented-

● Advance, accelerate, or compact content
● Support abstract and advanced higher-level thinking through differentiated questioning
● Allowance for adapting of content for individual student interests
● Provide assignments geared to development in areas of affect, creativity, cognition, and research skills

504-

Modifications and accommodations must be aligned to the 504 plan and uphold expectations of the plan lawfully.

● Provide additional time for in-class assignments and assessments
● Allow for computer with voice output, spelling and grammar checker
● Offer individual help throughout the lesson
● Modify readings and primary source documents

At Risk-

● Provide small group instruction for identified areas for growth
● Allow for alternate pathways for completion of assignments.
● Provide additional wait time for student responses to questions
● Pair oral instruction with visuals
● Utilize manipulatives

For a full list of Accommodations and Modifications, please click here.

21ST CENTURY LIFE AND CAREER STANDARDS

- ❏ Apply appropriate academic and technical skills.
- ❏ Communicate clearly and effectively and with reason.
- ❏ Demonstrate creativity and innovation.
- ❏ Employ valid and reliable research strategies.
- ❏ Utilize critical thinking to make sense of problems and persevere in solving them.
- ❏ Model integrity, ethical leadership, and effective management.
- ❏ Plan education and career paths aligned to personal goals.
- ❏ Use technology to enhance productivity.

LINKS TO CAREERS: https://career.ku.edu/computer-science

Unit 4

Data Collection

Summary and Rationale

This unit introduces the data structures array, ArrayList, and 2D array, which are used to represent collections of related data using a single variable rather than multiple variables. 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. An 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 dataset grows, such as when using a large real-world dataset. A 2D array is most suitable to represent a table. 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 discuss additional traversal patterns, such as column-major or back-and-forth. Just as there are useful standard algorithms when dealing with primitive data, there are standard algorithms to use with data structures. Additional algorithms, such as two common searching and three common sorting algorithms, are also covered.

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 concept. In this unit, students will learn how to write simple recursive methods and determine the purpose or output of a recursive method by tracing. Students will revisit how control is passed when methods are called, which is necessary knowledge when working with recursion. Tracing skills are also helpful.

Also in this unit, students will learn about privacy concerns related to storing large amounts of personal data and about what can happen if such information is compromised. They will also examine the potential for bias in collecting and using data.

Recommended Pacing

For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)	
AP - Algorithms & Programming	
2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.
DA - Data and Analysis	
2-DA-08	Collect data using computational tools and transform the data to make it more useful and reliable.
IC - Impacts of Computing	
2-IC-21	Evaluate the cultural and social implications of computing.
AP Computer Science A Conceptual Framework	
DAT-1	Understanding arrays, ArrayLists, and 2D arrays as ways to represent collections of data.
ALG-1	Apply common algorithmic strategies using iteration, including searching and sorting algorithms.
MOD-1	Using methods to organize and manipulate data structures effectively.

REC-1	Implementing recursive solutions for searching and sorting algorithms.
IO-1	Reading from and writing to text files to handle persistent data.
ETC-1	Recognizing the impact of data collection, including privacy and bias concerns.
CSA Topics	
4.1	Ethical and Social Issues Around Data Collection
4.2	Introduction to Using Data Sets
4.3	Array Creation and Access
4.4	Array Traversals
4.5	Implementing Array Algorithms
4.6	Using Text Files
4.7	Wrapper Classes
4.8	ArrayList Methods
4.9	ArrayList Traversal
4.10	Implementing ArrayList Algorithms
4.11	2D Array Creation and Access
4.12	2D Array Traversals
4.13	Implementing 2D Array Algorithms
4.14	Searching Algorithms
4.15	Sorting Algorithms
4.16	Recursion
4.17	Recursive Searching and Sorting
Interdisciplinary Connections	
Standard	
6.1.12.CivicsPI.5	Analyze how ethical decisions impact individuals and society.
HS-LS-ETS1-3	Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs, including data analysis.
MA.K-12.6.1.12.C.3	Use statistical methods to analyze data and make decisions.

Enduring Understandings:	**Essential Questions:**
Understanding how to effectively store and manipulate collections of data is fundamental to building efficient and scalable programs. As programs grow in complexity, managing data becomes essential to ensure clarity, performance, and functionality. Different data structures, such as arrays, ArrayLists, and 2D arrays, offer specific advantages and limitations. Choosing the appropriate structure depends on the size, type, and behavior of the data being managed. Algorithms like searching and sorting are foundational tools for working with data collections. Knowing how and when to apply these algorithms allows developers to write programs that are both efficient and reliable. Recursion provides a powerful alternative to iteration by solving problems through self-referencing solutions. It encourages thinking about problems in terms of smaller, repeated patterns and helps develop abstract problem-solving skills. As students work with real-world data, they must also consider the ethical and societal impacts of computing. This includes understanding privacy concerns, recognizing bias in data, and ensuring responsible handling of information.	What type of personal data is being collected when using a food delivery app, and how does the app collect the data? How can programs leverage volcano data to make predictions about the next eruption? Why is an ArrayList more appropriate for storing your music playlist, while an array is more appropriate for storing your class schedule? What are some real world processes that are recursive in nature?
Evidence of Learning (Assessments)	
Tests Quizzes Homework Programs Projects / Labs Class participation	
Objectives (SLO)	

Students will know:	Students will be able to:
● How to create and use arrays, ArrayLists, and 2D arrays ● When to use arrays vs. ArrayLists ● How to traverse and manipulate data collections ● Basic searching and sorting algorithms ● How to read and write data using text files ● What wrapper classes are and how they're used ● How to write and trace simple recursive methods ● Ethical concerns related to data collection and privacy	● Explain ethical concerns related to data bias and collection ● Analyze and process data sets to inform algorithm design ● Write algorithms to search, modify, and analyze values in 1D and 2D arrays ● Declare, initialize, and traverse arrays using while, for, and enhanced for loops ● Use and manipulate elements in arrays of primitive types and objects ● Read and process data from text files using the Scanner and File classes ● Use wrapper classes (Integer, Double) to convert and manipulate data ● Conduct code reviews to refine and improve solutions collaboratively

Suggested Resources/Technology Tools

-Programming compiler codeHS
-Online notebook/journal
-AP CSA Practice Tests and Released Items
-AP Classroom
-Online Textbook/Practice Workbooks

Modifications

Special Education-
Modifications and accommodations must be aligned to students' IEPs and uphold expectations of the plan lawfully.
● Allow additional time for skill mastery.
● Provide small group/ individualized instruction.
● Allow use of assistive technology.
● Anticipate needs; pre-teach if needed.
● Have student repeat directions to check for understanding.
● Highlight text/ visual presentation.
● Modify assessment/assignment content, format, length (benchmarks, tests and quizzes) .
ML-
● Provide additional wait time for student responses to questions
● Provide simplification of sentence structure and repetition of questions/sentences exactly as stated before trying to rephrase.
● Speak clearly, using a normal tone and rate of speed, or slightly slower.
● Use directed reading activities. Ensure preview of text before assigned/read, provide pre-reading
Gifted and Talented-
● Advance, accelerate, or compact content
● Support abstract and advanced higher-level thinking through differentiated questioning
● Allowance for adapting of content for individual student interests
● Provide assignments geared to development in areas of affect, creativity, cognition, and research skills
504-
Modifications and accommodations must be aligned to the 504 plan and uphold expectations of the plan lawfully.
● Provide additional time for in-class assignments and assessments
● Allow for computer with voice output, spelling and grammar checker

● Offer individual help throughout the lesson
● Modify readings and primary source documents

At Risk-

● Provide small group instruction for identified areas for growth
● Allow for alternate pathways for completion of assignments.
● Provide additional wait time for student responses to questions
● Pair oral instruction with visuals
● Utilize manipulatives

For a full list of Accommodations and Modifications, please click here.

21ST CENTURY LIFE AND CAREER STANDARDS

- ❏ Apply appropriate academic and technical skills.
- ❏ Communicate clearly and effectively and with reason.
- ❏ Demonstrate creativity and innovation.
- ❏ Employ valid and reliable research strategies.
- ❏ Utilize critical thinking to make sense of problems and persevere in solving them.
- ❏ Model integrity, ethical leadership, and effective management.
- ❏ Plan education and career paths aligned to personal goals.
- ❏ Use technology to enhance productivity.

LINKS TO CAREERS: https://career.ku.edu/computer-science