PISCATAWAY TOWNSHIP SCHOOLS

Dr. Frank Ranelli
Superintendent of Schools

Dr. William Baskerville
Assistant Superintendent

AP Computer Science A

Content Area: **Mathematics**

Grade Span: **10-12**

Revised by: **Brinder Soin**

Title **APCS A Teacher**

Approval Date: **August 2021**

Piscataway Township Schools
1515 Stelton Road
Piscataway, NJ 08854-1332
732 572-2289, ext. 2561
Fax 732 572-1540
www.piscatawayschools.org

COURSE OVERVIEW

Description

This is a college level computer science course , and is conducted in a rigorous manner. It is designed for the students who have an interest in pursuing computer science in college and have a strong background in mathematics.

The course emphasizes proper programming methodology, algorithmic development, data structures, and object oriented programming (OOP) and design. Students will become proficient in creating and implementing classes. This includes creating new classes using inheritance and implementing interfaces.

Topics studied in this course include a review of computer platforms and architecture; objects and primitive data; program statements; writing and enhancing classes; inheritance; one and two-dimensional arrays; recursion; and searching and sorting techniques. A detailed scope and sequence is shown below.

Every area that is referenced in the AP Computer Science A Course Description is studied. Students will be expected to use their knowledge of proper programming and design techniques when creating programs in this course. Programs are implemented using the JAVA programming language. Upon completion of the course, students will be able to take the College Board AP Computer Science A exam.

All AP Computer Science classes are taught in a computer lab. Each student has a computer workstation. Lessons are taught using Socratic methods. Portions of programs are presented and discussed. Students create sample classes with the instructor.

Goals

- ➔ The course teaches students to design and implement computer based solutions to problems in a variety of application areas.
- ➔ The course teaches students to use and implement commonly used algorithms and data structures.
- ➔ The course teaches students to develop and select appropriate algorithms and data structures to solve problems.
- ➔ The course teaches students to code fluently in an object- oriented paradigm using the programming language Java. The course teaches students to use standard Java library classes from the AP Java subset.
- ➔ The course teaches students to identify the major hardware and software components of a computer system, their relationship to one another, and the roles of these components within the system.
- ➔ The course teaches students to read and understand a large program consisting of several classes and interacting objects, and enables students to read and understand the current AP Computer Science list of assigned labs posted on AP Central.
- ➔ The course teaches students to recognize the ethical and social implications of computer use.

Scope and Sequence

Unit	Topic	Length
Unit 1	Primitive Types	5 Days
Unit 2	Using Objects	7 Days
Unit 3	Boolean Expressions and if Statements	6 Days
Unit 4	Iteration	8 Days
Unit 5	Writing Classes	9 Days
Unit 6	Array	6 Days
Unit 7	ArrayList	6 Days
Unit 8	2D Array	6 Days
Unit 9	Inheritance	8 Days
Unit 10	Recursion	2 Days

Resources

Text used:
J. Lewis, W. Loftus, and C. Cocking, Java Software Solutions for AP Computer Science, 1st ed., Prentice Hall, 2004.
Platforms used:
Java API - https://docs.oracle.com/en/java/javase/11/docs/api/index.html
IDE - www.eclipse.org
Online IDE - www.codiva.io

UNIT 1: Primitive Types

Summary and Rationale

This unit introduces students to the Java programming language and the use of classes, providing students with a firm foundation of concepts that will be leveraged and built upon in all future units. Students will focus on writing the main method and will start to call pre existing methods to produce output. Students will start to learn about three built-in data types and learn how to create variables, store values, and interact with those variables using basic operations. The ability to write expressions is essential to representing the variability of the real world in a program and will be used in all future units. Primitive data is one of two categories of variables covered in this course. The other category, reference data, will be covered in Unit 2.

Recommended Pacing

5 days (1 day = 80 minutes)

State Standards

Core Idea: A computing system involves interaction among the user, hardware, application software, and system software.

- ➔ 8.1.12.CS.2: Model interactions between application software, system software, and hardware.
- ➔ 8.1.12.CS.3: Compare the functions of application software, system software, and hardware.

Core Idea: Choices individuals make about how and cost, speed, reliability, accessibility, privacy, and integrity.

- ➔ 8.1.12.DA.2: Describe the trade-offs in how and where data is organized and stored.
- ➔ 8.1.12.DA.3: Translate between decimal numbers and binary numbers.
- ➔ 8.1.12.DA.4: Explain the relationship between binary numbers and the storage and use of data in a computing device.

Instructional Focus

Unit Enduring Understandings

MOD-1: Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence.

VAR-1: To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.

CON-1: The way variables and operators are sequenced and combined in an expression determines the computed result.

CON-1: The way variables and operators are sequenced and combined in an expression determines the computed result.

CON-1: The way variables and operators are sequenced and combined in an expression determines the computed result.

Unit Essential Questions

- A computer system consists of hardware and software that work together to help us solve problems.
- To execute a program, the computer first copies the program from secondary memory to main memory. The CPU then reads the program instruc- tions from main memory, executing them one at a time until the program ends.
- The operating system provides a user interface and manages computer resources.
- Digital computers store information by breaking it into pieces and representing each piece as a number.
- Binary values are used to store all information in a computer because the devices that store and manipulate binary information are inexpensive and reliable.
- There are exactly 2^N permutations of N bits. Therefore N bits can represent up to 2^N unique items.
- An address is a unique number associated with each memory location. It is used when storing and retrieving data from memory.
- Data written to a memory location overwrites and destroys any information that was previously stored at that location. Data read from a memory location leaves the value in memory unaffected.
- Main memory is volatile, meaning the stored information is maintained only as long as electric power is supplied. Secondary memory devices are usually nonvolatile.
- The von Neumann architecture and the fetch-decode-execute cycle form the foundation of computer processing.
- The speed of the system clock indicates how fast the CPU executes instructions.
- The purpose of writing a program is to solve a problem.
- The first solution we design to solve a problem may not be the best one.
- Comments do not affect a program's processing; instead, they serve to facilitate human comprehension.
- The main method must always be defined using the words public, static, and void.
- Inline documentation should provide insight into your code. It should not be ambiguous or belabor the obvious.
- Java is case sensitive. The uppercase and lowercase versions of a letter are distinct. You should use a consistent case convention for different types of identifiers.
- Identifier names should be descriptive and readable.
- Appropriate use of white space makes a program easier to read and understand.
- A Java compiler translates Java source code into Java bytecode. A Java interpreter translates and executes the bytecode.
- Java is architecture neutral because Java bytecode is not associated with any particular hardware platform.
- The syntax rules of a programming language dictate the form of a program. The semantics dictate the meaning of the program statements.
- A computer follows our instructions exactly. The programmer is responsible for the accuracy and reliability of a program.
- A Java program must be syntactically correct or the compiler will not produce bytecode.
- The information we manage in a Java program is either represented as primitive data or as objects.
- An abstraction hides details. A good abstraction hides the right details at the right time so that we can manage complexity.

- A variable is a name for a memory location used to hold a value of a particular data type.
- A variable can store only one value of its declared type.
- Java is a strongly typed language. Each variable is associated with a specific type for the duration of its existence, and we cannot assign a value of one type to a variable of an incompatible type.
- Constants are similar to variables, but they hold a particular value for the duration of their existence.
- Java has two kinds of numeric values: integers and floating point. There are four integer data types (byte, short, int, and long) and two floating point data types (float and double).

Objectives

Students will know and be able to:

- Call System class methods to generate output to the console.
- Create string literals.
- Identify the most appropriate data type category for a particular specification.
- Declare variables of the correct types to represent primitive data.
- Evaluate arithmetic expressions in a program code.
- Evaluate what is stored in a variable as a result of an expression with an assignment statement.
- Evaluate arithmetic expressions that use casting.

Resources

Text used:
J. Lewis, W. Loftus, and C. Cocking, Java Software Solutions for AP Computer Science, 1st ed., Prentice Hall, 2004.
Platforms used:
Java API - https://docs.oracle.com/en/java/javase/11/docs/api/index.html
IDE - www.eclipse.org
Online IDE - www.codiva.io

UNIT 2: Using Objects

Summary and Rationale

In the first unit, students used primitive types to represent real-world data and determined how to use them in arithmetic expressions to solve problems. 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 but will be very important in Units 5 and 9 when teaching students how to write their own classes and about inheritance relationships.

Recommended Pacing

7-days

State Standards

Core Idea: Engineering design is a complex process in which creativity, content knowledge, research, and analysis are used to address local and global problems.
Decisions on trade-offs involve systematic comparisons of all costs and benefits, and final steps that may involve redesigning for optimization.

- ➔ 8.2.12.ED.1: Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.
- ➔ 8.2.12.ED.2: Create scaled engineering drawings for a new product or system and make modifications to increase optimization based on feedback.
- ➔ 8.2.12.ED.3: Evaluate several models of the same type of product and make recommendations for a new design based on a cost benefit analysis.
- ➔ 8.2.12.ED.4: Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.

Instructional Focus

Unit Enduring Understandings

MOD-1: Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence.

VAR-1: To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.

CON-1: The way variables and operators are sequenced and combined in an expression determines the computed result.

Unit Essential Questions

- Many programming statements involve expressions. Expressions are combinations of one or more operands and the operators used to perform a calculation.
- Java follows a well-defined set of rules that govern the order in which operators will be evaluated in an expression. These rules form an operator precedence hierarchy.
- Avoid narrowing conversions because they can lose information.
- The new operator returns a reference to a newly created object.
- The Java standard class library is a useful set of classes that anyone can use when writing Java programs.
- A package is a Java language element used to group related classes under a common name.
- The Keyboard class is not part of the Java standard library. It is therefore not available on all Java development platforms.

Objectives

Students will know and be able to:

- Explain the relationship between a class and an object.
- Identify, using its signature, the correct constructor being called.
- For creating objects:
  - Create objects by calling constructors without parameters.
  - Create objects by calling constructors with parameters.
- Define variables of the correct types to represent reference data.
- Call non-static void methods without parameters.
- Call non-static void methods with parameters.
- Call non-static non-void methods with or without parameters.
- For String class:
  - Create String objects.
  - Call String methods.
- For wrapper classes:
  - Create Integer objects.
  - Call Integer methods.
  - Create Double objects.
  - Call Double methods.
- Call static methods.
- Evaluate expressions that use the Math class methods.

Resources

Text used:
J. Lewis, W. Loftus, and C. Cocking, Java Software Solutions for AP Computer Science, 1st ed., Prentice Hall, 2004.
Platforms used:
Java API - https://docs.oracle.com/en/java/javase/11/docs/api/index.html
IDE - www.eclipse.org
Online IDE - www.codiva.io

UNIT 3: Boolean Expressions and if Statements

Summary and Rationale

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.

The third building block of all algorithms is iteration, which you will cover in Unit 4. Selection and iteration work together to solve problems.

Recommended Pacing

6 days

State Standards

Core Idea: Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.

- ➔ 8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.

Core Idea: Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.

- ➔ 8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.

Core Idea: Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.

- ➔ 8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
- ➔ 8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.

Instructional Focus

Unit Enduring Understandings

CON-1: The way variables and operators are sequenced and combined in an expression determines the computed result.

CON-2: Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

Unit Essential Questions

- Software requirements specify what a program must accomplish.
- A software design specifies how a program will accomplish its requirements.
- An algorithm is a step-by-step process for solving a problem, often expressed in pseudocode.
- Implementation should be the least creative of all development activities.
- The goal of testing is to find errors. We can never really be sure that all errors have been found.
- Conditionals and loops allow us to control the flow of execution through a method.
- An if statement allows a program to choose whether to execute a particular statement.
- Even though the compiler does not care about indentation, proper inden- tation is important for human readability; it shows the relationship between one statement and another.
- An if-else statement allows a program to do one thing if a condition is true and another thing if the condition is false.
- In a nested if statement, an else clause is matched to the closest unmatched if.
- A break statement is usually used at the end of each case as an alternative of a switch statement to jump to the end of the switch.
- A switch statement could be implemented as a series of if-else state- ments, but the switch is sometimes a more convenient and readable construct.

Objectives

Students will know and be able to:

- Evaluate Boolean expressions that use relational operators in program code.
- Represent branching logical processes by using conditional statements.
- Represent branching logical processes by using nested conditional statements.
- Evaluate compound Boolean expressions in program code.
- Compare and contrast equivalent Boolean expressions.
- Compare object references using Boolean expressions in program code.

Resources

Text used:
J. Lewis, W. Loftus, and C. Cocking, Java Software Solutions for AP Computer Science, 1st ed., Prentice Hall, 2004.
Platforms used:
Java API - https://docs.oracle.com/en/java/javase/11/docs/api/index.html
IDE - www.eclipse.org
Online IDE - www.codiva.io

UNIT 4: Iteration

Summary and Rationale

This unit focuses on iteration using while and for loops. As you saw in Unit 3, Boolean expressions are useful when a program needs to perform different operations under different conditions. Boolean expressions are also one of the main components in iteration. 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. In addition, it is a necessary component of several standard algorithms, including searching and sorting, which will be covered in later units.

Recommended Pacing

8 days

State Standards

Core Idea: Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.

- ➔ 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- ➔ 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

Core Idea: Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.

- ➔ 8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.

Instructional Focus

Unit Enduring Understandings

CON-2: Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

Unit Essential Questions

- Logical operators return a boolean value and are often used to construct sophisticated conditions.
- The relative order of characters in Java is defined by the Unicode character set.
- The compareTo method can be used to determine the relative order of strings. It determines lexicographic order, which does not correspond exactly to alphabetical order.
- The prefix and postfix increment and decrement operators have subtle effects on programs because of differences in when they are evaluated.
- A while statement allows a program to execute the same statement multi- ple times.
- We must design our programs carefully to avoid infinite loops. The body of the loop must eventually make the loop condition false.
- A do statement executes its loop body at least once.

- A for statement is usually used when a loop will be executed a set number of times.

Objectives

Students will know and be able to:

- Represent iterative processes using a while loop.
- For algorithms in the context of a particular specification that does not require the use of traversals:
  - Identify standard algorithms.
  - Modify standard algorithms.
  - Develop an algorithm.
- Represent iterative processes using a for loop.
- For algorithms in the context of a particular specification that involves String objects:
  - Identify standard algorithms.
  - Modify standard algorithms.
  - Develop an algorithm.
- Represent nested iterative processes.
- Compute statement execution counts and informal run-time comparison of iterative statements.

Resources

Text used:
J. Lewis, W. Loftus, and C. Cocking, Java Software Solutions for AP Computer Science, 1st ed., Prentice Hall, 2004.
Platforms used:
Java API - https://docs.oracle.com/en/java/javase/11/docs/api/index.html
IDE - www.eclipse.org
Online IDE - www.codiva.io

Unit 5: Writing Classes

Summary and Rationale

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. Students will build on what they learn in this unit to represent relationships between classes through hierarchies, which appear in Unit 9.

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.

Recommended Pacing

9 days

State Standards

Core Idea: Engineering design is a complex process in which creativity, content knowledge, research, and analysis are used to address local and global problems.
Decisions on trade-offs involve systematic comparisons of all costs and benefits, and final steps that may involve redesigning for optimization.

- ➔ 8.2.12.ED.1: Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.
- ➔ 8.2.12.ED.2: Create scaled engineering drawings for a new product or system and make modifications to increase optimization based on feedback.
- ➔ 8.2.12.ED.3: Evaluate several models of the same type of product and make recommendations for a new design based on a cost benefit analysis.
- ➔ 8.2.12.ED.4: Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.

Core Idea: Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.

- ➔ 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- ➔ 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

Instructional Focus

Unit Enduring Understandings

MOD-2: Programmers use code to represent a physical object or nonphysical concept, real or imagined, by defining a class based on the attributes and/or behaviors of the object or concept.

MOD-3: When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

VAR-1: To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.

IOC-1: While programs are typically designed to achieve a specific purpose, they may have unintended consequences.

Unit Essential Questions

- What is the difference between an object and a class?
- What is the scope of a variable?
- What are UML diagrams designed to do?
- Objects should be self-governing. Explain.
- What is a modifier?
- Describe each of the following:
  - public method
  - private method
  - public variable
  - private variable
- What does the return statement do?
- Explain the difference between an actual parameter and a formal parameter.
- What are constructors used for? How are they defined?
- How are overloaded methods distinguished from each other?
- What is method decomposition?
- Explain how a class can have an association with itself.
- What is an aggregate object?

Objectives

Students will know and be able to:

- Designate access and visibility constraints to classes, data, constructors, and methods.
- Designate private visibility of instance variables to encapsulate the attributes of an object
- Define instance variables for the attributes to be initialized through the constructors of a class.
- Describe the functionality and use of program code through comments.
- Define behaviors of an object through non-void methods without parameters written in a class.
- Define behaviors of an object through void methods with or without parameters written in a class.
- Define behaviors of an object through non-void methods with parameters written in a class.
- Define behaviors of a class through static methods.
- Define the static variables that belong to the class.
- Explain where variables can be used in the program code.

- Evaluate object reference expressions that use the keyword this.
- Explain the ethical and social implications of computing systems.

Resources

Text used:
J. Lewis, W. Loftus, and C. Cocking, Java Software Solutions for AP Computer Science, 1st ed., Prentice Hall, 2004.
Platforms used:
Java API - https://docs.oracle.com/en/java/javase/11/docs/api/index.html
IDE - www.eclipse.org
Online IDE - www.codiva.io

Unit 6: Array

Summary and Rationale

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, we apply standard algorithms to arrays; however, these same algorithms are used with ArrayLists and 2D arrays as well. Additional standard algorithms, such as standard searching and sorting algorithms, will be covered in the next unit.

Recommended Pacing

8 days

State Standards

Core Idea: Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information.

- ➔ 8.1.12.DA.5: Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.

Core Idea: The accuracy of predictions or inferences made from a computer model is affected by the amount, quality, and diversity of data.

- ➔ 8.1.12.DA.6: Create and refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.

Core Idea: The usability, dependability, security, and accessibility of devices within integrated systems are important considerations in their design as they evolve.

- ➔ 8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.

Instructional Focus

Unit Enduring Understandings

VAR-2: To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

CON-2: Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

Unit Essential Questions

- Explain the concept of array bounds checking. What happens when a Java array is indexed with an invalid value?
- Describe the process of creating an array. When is memory allocated for the array?
- What is an off-by-one error? How does it relate to arrays?
- What does an array initializer list accomplish?
- Can an entire array be passed as a parameter? How is this accom- plished?
- How is an array of objects created?
- What is a command-line argument?
- What are parallel arrays?
- Which is better: selection sort or insertion sort? Explain.

Objectives
Students will know and be able to: ● Represent collections of related primitive or object reference data using one-dimensional (1D) array objects. ● Traverse the elements in a 1D array. ● Traverse the elements in a 1D array object using an enhanced for loop. ● For algorithms in the context of a particular specification that requires the use of array traversals: ○ Identify standard algorithms. ○ Modify standard algorithms. ○ Develop an algorithm.

Resources
Text used: **J. Lewis, W. Loftus, and C. Cocking, Java Software Solutions for AP Computer Science, 1st ed., Prentice Hall, 2004.** **Platforms used:** **Java API - https://docs.oracle.com/en/java/javase/11/docs/api/index.html** **IDE - www.eclipse.org** **Online IDE - www.codiva.io**

Unit 7: ArrayList

Summary and Rationale
As 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.

Recommended Pacing

6 days

State Standards

Core Idea: Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information.

- ➔ 8.1.12.DA.5: Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.

Core Idea: The accuracy of predictions or inferences made from a computer model is affected by the amount, quality, and diversity of data.

- ➔ 8.1.12.DA.6: Create and refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.

Core Idea: Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks

- ➔ 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- ➔ 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

Instructional Focus

Unit Enduring Understandings

VAR-2: To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

CON-2: Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

IOC-1: While programs are typically designed to achieve a specific purpose, they may have unintended consequences.

Unit Essential Questions

- How to declare and import an ArrayList.
- How to initialize an ArrayList?
- List and explain the most important functions of the ArrayList class.
- What are the advantages of using an ArrayList object as opposed to an array? What are the disadvantages?

Objectives

Students will know and be able to:

- Represent collections of related object reference data using ArrayList objects.
- For ArrayList objects:
  - Traverse using a for or while loop
  - Traverse using an enhanced for loop
- For algorithms in the context of a particular specification that requires the use of ArrayList traversals:
  - Identify standard algorithms.
  - Modify standard algorithms.
  - Develop an algorithm.
- Apply sequential / linear search algorithms to search for specific information in array or ArrayList objects.
- Apply selection sort and insertion sort algorithms to sort the elements of array or ArrayList objects.
- Compute statement execution counts and informal run-time comparison of sorting algorithms.
- Explain the risks to privacy from collecting and storing personal data on computer systems.

Resources

Text used:
J. Lewis, W. Loftus, and C. Cocking, Java Software Solutions for AP Computer Science, 1st ed., Prentice Hall, 2004.
Platforms used:
Java API - https://docs.oracle.com/en/java/javase/11/docs/api/index.html
IDE - www.eclipse.org
Online IDE - www.codiva.io

Unit 8: 2D Array

Summary and Rationale

In Unit 6, students learned how 1D arrays store large amounts of related data. These same 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.

Recommended Pacing

6 days

State Standards

Core Idea: Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information.

- ➔ 8.1.12.DA.5: Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.

Core Idea: The accuracy of predictions or inferences made from a computer model is affected by the amount, quality, and diversity of data.

- ➔ 8.1.12.DA.6: Create and refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.

Core Idea: Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks

- ➔ 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- ➔ 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

Instructional Focus

Unit Enduring Understandings

VAR-2: To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

CON-2: Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

Unit Essential Questions

- How to declare a 2D array?
- What are different ways to initialize a 2D array?
- How to traverse a 2D array in row major order.
- How to traverse a 2D array in column major order.
- How to store objects in a 2D array?
- How to access and manipulate objects stored in an 2D array.

Objectives

Students will know and be able to::

- Represent collections of related primitive or object reference data using two-dimensional (2D) array objects.
- For 2D array objects:
  - Traverse using nested for loops.
  - Traverse using nested enhanced for loops.
- For algorithms in the context of a particular specification that requires the use of 2D array traversals:
  - Identify standard algorithms.
  - Modify standard algorithms.
  - Develop an algorithm.

Resources

Text used:
J. Lewis, W. Loftus, and C. Cocking, Java Software Solutions for AP Computer Science, 1st ed., Prentice Hall, 2004.
Platforms used:
Java API - https://docs.oracle.com/en/java/javase/11/docs/api/index.html
IDE - www.eclipse.org
Online IDE - www.codiva.io

Unit 9: Inheritance

Summary and Rationale

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.

Recommended Pacing

8 days

State Standards

Core Idea: Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences.

- ➔ 8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

Core Idea: Complex programs are developed, tested, and analyzed by teams drawing on the members' diverse strengths using a variety of resources, libraries, and tools.

- ➔ 8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
- ➔ 8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.
- ➔ 8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.

Instructional Focus

Unit Enduring Understandings

MOD-3: When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

Unit Essential Questions

- Describe the relationship between a parent class and a child class.
- How does inheritance support software reuse?
- What relationship should every class derivation represent?
- Why would a child class override one or more of the methods of its parent class?
- Why is the super reference important to a child class?
- What is the significance of the Object class?
- What is the role of an abstract class?
- Are all members of a parent class inherited by the child? Explain.
- What is polymorphism?
- How does inheritance support polymorphism?
- How is overriding related to polymorphism?

- What is an interface hierarchy?
- How can polymorphism be accomplished using interfaces?
- What is an adapter class?

Objectives

Students will know and be able to::

- Create an inheritance relationship from a subclass to the superclass.
- Define reference variables of a superclass to be assigned to an object of a subclass in the same hierarchy.
- Call methods in an inheritance relationship.
- Call Object class methods through inheritance.

Resources

Text used:
J. Lewis, W. Loftus, and C. Cocking, Java Software Solutions for AP Computer Science, 1st ed., Prentice Hall, 2004.
Platforms used:
Java API - https://docs.oracle.com/en/java/javase/11/docs/api/index.html
IDE - www.eclipse.org
Online IDE - www.codiva.io

Unit 10: Recursion

Summary and Rationale

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. Tracing skills introduced in Unit 2 are helpful for determining the purpose or output of a recursive method. In this unit, students will learn how to write simple recursive methods and determine the purpose or output of a recursive method by tracing.

Recommended Pacing

3 days

State Standards

Core Idea: Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.

- ➔ 8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- ➔ 8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

Instructional Focus

Unit Enduring Understandings

CON-2: Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

Unit Essential Questions

- What is recursion?
- What is infinite recursion?
- When is a base case needed for recursive processing?
- Is recursion necessary?
- When should recursion be avoided?
- What is indirect recursion?
- Explain the general approach to solving the Towers of Hanoi puzzle. How does it relate to recursion?

Objectives

Students will know and be able to::

- Explain how computing innovations are developed by groups of people.
- Determine the result of executing recursive methods.
- Apply recursive search algorithms to information in String, 1D array, or ArrayList objects.

Resources

Platforms used:
Text used:
J. Lewis, W. Loftus, and C. Cocking, Java Software Solutions for AP Computer Science, 1st ed., Prentice Hall, 2004.
Platforms used:
Java API - https://docs.oracle.com/en/java/javase/11/docs/api/index.html
IDE - www.eclipse.org
Online IDE - www.codiva.io