

Southern York County School District Instructional Plan

Course/Subject: AP Computer Science A

Grade Level: 11, 12

Textbook(s)/Instructional Materials Used:

- **Multiple-Choice & Free-Response Questions in Preparation for the AP Computer Science Examination;** D&S Marketing Systems, Inc.; 9th edition (2015); ISBN: 978-1934780343
- **Student's Solutions Manual for Computer Science;** by Leon Schram (ISBN-13: 978-1934780350)

Dates: September

Unit Plan: 1

Stage 1 – Desired Results

PA Standard(s)/Assessment Anchors Addressed:

15.4.12.H Use programming languages to develop logical thinking and problem solving skills.

15.4.8.I Compare and contrast programming languages; select most appropriate one to complete a specific task.

3.4.12.C2 Apply the concept that engineering design is influenced by personal characteristics, such as creativity, resourcefulness, and the ability to visualize and think abstractly.

3.4.12.C3 Apply the concept that many technological problems require a multi-disciplinary approach.

Understanding(s):

Students will understand

1. Computer Science is integrated into many disciplines. (3.4.12.C3)
2. The difference between hardware and software. (15.4.8.I)
3. The difference between low-level programming languages and high-level languages. (15.4.8.I)
4. The history of the Java programming language and its importance in industry (15.4.8.I)
5. Problem solving is a planned event. (3.4.12.C2)
6. Computer Languages must follow a specific syntax. (15.4.12.H)

Essential Question(s):

- How does the study of Computer Science help you to be a better problem solver?
- Why is Java an important program language?
- How are a Computer Language and Human Language similar?
- How do hardware and software make up computer architecture?
- How are the fundamental concepts of object-oriented programming used?
- How do we edit, compile and run a program using a java development environment?

Learning Objectives:

Students will know...

- How hardware and software make up computer architecture.(15.4.8.I)
- The binary representation of data and programs in computers.(3.4.12.C3)
- The evolution of programming languages.(15.4.8.I)
- The software development process.(15.4.8.I)
- The fundamental concepts of object-oriented programming. (15.4.12.H)
- Why Java is an important programming language.(15.4.8.I)
- The Java virtual machine and byte code.(15.4.8.I)
- The different user interface style. (15.4.8.I)

Students will be able to:

- Define what Computer Science is and is not.
- Create a timeline for the history of computers.
- List hardware and software components.
- Write whether a computer language is high-level or low-level.
- Write the binary equivalent of a base 10 numeral.
- Explain the JVM.
- Use the Bluejay interface environment for programming.
- Write a simple program using the Bluejay environment
- Write a program using the turtle graphics package.
- Construct and use numeric and string literals.

<ul style="list-style-type: none"> ▪ The structure of a simple Java program. (3.4.12.C2) ▪ How to write a simple program. (3.4.12.C2) ▪ How to edit, compile, and run a program using a Java development environment. (15.4.12.H) ▪ How to construct and use numeric and string literals. (3.4.12.C2) ▪ How to name and use variables and constants.(3.4.12.C2) ▪ How to create arithmetic expressions. (3.4.12.C3) ▪ The precedence of different arithmetic operators. (3.4.12.C3) ▪ How to concatenate two strings or a number and a string. (15.4.12.H) ▪ How and when to use comments in a program. (15.4.12.H) ▪ The difference between syntax errors, run-time errors, and logic errors. (3.4.12.C2) ▪ How to insert output statements to debug a program. (3.4.12.C2) 	<ul style="list-style-type: none"> ▪ Name and use variables and constants. ▪ Create arithmetic expressions. ▪ List the precedence of operators. ▪ Concatenate two strings or a number and a string. ▪ Comment a program correctly. ▪ Identify errors as syntax, logical, or runtime.
---	--

Dates: October	Unit Plan: 2
-----------------------	---------------------

Stage 1 – Desired Results

PA Standard(s)/Assessment Anchors Addressed:
15.4.12.H Use programming languages to develop logical thinking and problem solving skills.
15.4.8.J Create a complex computer program to solve a problem
3.4.12.C2 Apply the concept that engineering design is influenced by personal characteristics, such as creativity, resourcefulness, and the ability to visualize and think abstractly.
CC.2.4.HS.B1 Summarize, represent, and interpret data on a single count or measurement variable.
CC.2.2.HS.D.2 Write expressions in equivalent forms to solve problems.

<p>Understanding(s): Students will understand...</p> <ol style="list-style-type: none"> 1. The math class provided by java will allow the use of many mathematical operations. (CC.2.4.HS.B1) 2. Decision statements and looping structures give the programmer the ability to create powerful programs and eliminate redundancy. (15.4.8.J) 3. Object Oriented Programming and the designing of classes will encourage the use of private and public methods. (3.4.12.C2) 4. Classes and methods are used to organize programs into separate parts that complete different tasks. (15.4.8.J) 	<p>Essential Question(s):</p> <ul style="list-style-type: none"> ▪ Why is it important for computer languages to have the ability to make decisions? ▪ How is Object Oriented Programming different than procedural programming? ▪ How can we design and organize a program in terms of a view class and model class?
--	---

Learning Objectives:

<p>Students will know...</p> <ul style="list-style-type: none"> ▪ How to use the increment and decrement operators. (CC.2.2.HS.D.2). ▪ How to use standard math methods. (CC.2.4.HS.B1) ▪ How to use if and if-else statements to make choices. (15.4.8.J) ▪ How to use while and for loops to repeat a process. (15.4.8.J) ▪ How to construct appropriate conditions for control statements using relational operators. (3.4.12.C2) ▪ How to detect and correct common errors involving loops. (3.4.12.C2) ▪ How to design and implement a simple class from user requirements. (15.4.12.H) ▪ How to organize a program in terms of a view class and a model class (15.4.12.H) ▪ How to use visibility modifiers to make methods visible to clients and restrict access to data within a class. (15.4.8.J) ▪ How to write appropriate mutator methods, accessor methods, and constructors for a class. (15.4.8.J) ▪ How parameters transmit data to methods. (15.4.8.J) ▪ How to use instance variables, local variables, and parameters appropriately. (15.4.12.H) ▪ How to organize a complex task in terms of helper methods. (15.4.12.H) 	<p>Students will be able to:</p> <ul style="list-style-type: none"> ▪ Use the increment and decrement operators. ▪ Use standard Math methods. ▪ Use if and if-else statements to make decisions. ▪ Recognize the differences between == and .isEqual() with Strings. ▪ Students understand boundary condition errors. ▪ Use while and for loops to repeat a process. ▪ Use Boolean variables and conditions to control while loops. ▪ Construct appropriate conditions for control statements using relational operators. ▪ Detect and correct common errors involving loops. ▪ Design and implement a simple class from user requirements. ▪ Organize a program in terms of a view class and a model class. ▪ Use visibility modifiers to make methods visible to clients and restrict access to data within a class. ▪ Write appropriate mutator methods, accessor methods, and constructors for a class. ▪ Understand how parameters transmit data to methods. ▪ Use instance variables, local variables, and parameters appropriately. ▪ Organize a complex task in terms of helper methods.
<p>Dates: November</p>	<p>Unit Plan: 3</p>
<p>Stage 1 – Desired Results</p>	
<p>PA Standard(s)/Assessment Anchors Addressed: 15.4.12.H Use programming languages to develop logical thinking and problem solving skills. 15.4.8.J Create a complex computer program to solve a problem 3.4.12.C2 Apply the concept that engineering design is influenced by personal characteristics, such as creativity, resourcefulness, and the ability to visualize and think abstractly. 3.4.12.C3 Apply the concept that many technological problems require a multi-disciplinary approach.</p>	
<p>Understanding(s): Students will understand...</p> <ol style="list-style-type: none"> 1. Boolean expressions can be used to allow programs to make complex decisions. (15.4.8.J) 2. All of software’s components are combined and tested as a single unit. (15.4.12.H) 3. The advantages of each type of loop. (15.4.8.J) 4. Nesting control statements and loops will enable a programmer to increase their ability to test for situations. (3.4.12.C3) 	<p>Essential Question(s):</p> <ul style="list-style-type: none"> ▪ How do we represent real-world concepts in software? ▪ How can software be created to model more sophisticated decisions? ▪ How do logical operators and nesting loops and decision statements allow a programmer to write more complex code? ▪ What part do these concepts play in the testing of programs?

<p>5. Logical errors can be found in nested if statements by testing the limits in the program. (3.4.12.C2)</p>	
<p>Learning Objectives: Students will know...</p> <ul style="list-style-type: none"> ▪ How to construct complex Boolean expressions. (15.4.8.J) ▪ How to construct truth tables for Boolean expressions. (15.4.12.H) ▪ The logic of nested if statements and extended if statements. (3.4.12.C3) ▪ How to test if statements in a comprehensive manner. (3.4.12.C2) ▪ How to construct nested loops. (3.4.12.C3) ▪ How to create appropriate test cases for if statements and loops. (3.4.12.C2) 	<p>Students will be able to:</p> <ul style="list-style-type: none"> ▪ Construct complex Boolean expressions using the logical operators and, or, and not. ▪ Use the proper syntax in a program that uses nested loops and complex Boolean expressions. ▪ Use the correct precedence while creating programs with if statements and loops. ▪ Construct truth tables for Boolean expressions. ▪ Describe key strategies in the test phase of a program. ▪ Implement nested if statements and extended if statements. ▪ Create programs that use both nested loops and complex Boolean expressions. ▪ Test if statements in a comprehensive manner. ▪ Construct nested loops. ▪ Create appropriate test cases for if statements and loops.
<p>Dates: December</p>	<p>Unit Plan: 4</p>
<p>Stage 1 – Desired Results</p>	
<p>PA Standard(s)/Assessment Anchors Addressed: 15.4.12.H Use programming languages to develop logical thinking and problem solving skills. 15.4.8.J Create a complex computer program to solve a problem 3.4.12.C2 Apply the concept that engineering design is influenced by personal characteristics, such as creativity, resourcefulness, and the ability to visualize and think abstractly. 3.4.12.C3 Apply the concept that many technological problems require a multi-disciplinary approach.</p>	
<p>Understanding(s): Students will understand...</p> <ol style="list-style-type: none"> 1. Using the GUI interface will improve the appearance of their program.(15.4.8.J) 2. A GUI uses different commands for different data types. (3.4.12.C3) 3. A GUI can be used to input and output data. (3.4.12.C2) 4. A GUI can activate different commands through the use of button objects. (15.4.12.H) 	<p>Essential Question(s):</p> <ul style="list-style-type: none"> ▪ Why is the appearance of a program’s output important to the users of the software? ▪ How can users interact with a computer and software? ▪ How can a program be made to look like software used every day? ▪ How can a GUI be used to improve a program?
<p>Learning Objectives: Students will know...</p> <ul style="list-style-type: none"> ▪ How to construct a query-driven, menu-driven, and graphical user interfaces. (3.4.12.C2) ▪ How to position window objects in a grid within a window. (15.4.12.H) ▪ How to write a method to handle users’ interactions with command buttons. (15.4.8.J) 	<p>Students will be able to:</p> <ul style="list-style-type: none"> ▪ Construct a query-driven terminal interface. ▪ Construct a menu-driven terminal interface. ▪ Construct programs with text boxes and buttons. ▪ Test a program appropriately that uses a GUI. ▪ Create GUIs of any size. ▪ Use Boolean expressions along with button commands to make a GUI more realistic. ▪ Construct a graphical user interface.

<ul style="list-style-type: none"> How to manipulate window objects to input and output integers, doubles, and text. (3.4.12.C3) 	<ul style="list-style-type: none"> Position window objects in a grid within a window. Write a method to handle users' interactions with command buttons. Construct a program with JTextField and JButton components. Manipulate window objects to input and output integers, doubles, and text.
---	---

Dates: January	Unit Plan: 5
-----------------------	---------------------

Stage 1 – Desired Results

PA Standard(s)/Assessment Anchors Addressed:
15.4.12.H Use programming languages to develop logical thinking and problem solving skills.
15.4.8.J Create a complex computer program to solve a problem
3.4.12.C2 Apply the concept that engineering design is influenced by personal characteristics, such as creativity, resourcefulness, and the ability to visualize and think abstractly.

<p>Understanding(s): <i>Students will understand...</i></p> <ol style="list-style-type: none"> Situations exist where a program manipulates large amounts of similar data. (15.4.8.J) Arrays have limitations and aren't always the best choice when writing a program. (3.4.12.C2) Arrays can be used in conjunction with loops to drastically shorten a programmer's code. (15.4.12.H) Situations exist that two-dimensional arrays are more useful because it allows similar information to stay together. (15.4.8.J) 	<p>Essential Question(s):</p> <ul style="list-style-type: none"> How do arrays enable the programmer to deal with large amounts of data? Why do arrays decrease the workload of the programmer? When is it appropriate to use a two-dimensional array over a single-dimensional array?
--	--

<p>Learning Objectives: <i>Students will know...</i></p> <ul style="list-style-type: none"> How to write programs that handle collections of similar items. (15.4.8.J) How to declare array variables and instantiate array objects. (15.4.8.J) How to manipulate arrays with loops. (15.4.12.H) How to write methods to manipulate arrays. (3.4.12.C2) How to create parallel arrays and two-dimensional arrays. (15.4.8.J) 	<p><i>Students will be able to:</i></p> <ul style="list-style-type: none"> Write programs that can handle collections of similar data. Determine the length of an array. Use subscripts with an array. Use arrays for the organization of data. Use arrays to manage data. Use arrays with objects and primitive data types. Declare array variables and instantiate array objects. Pass arrays through method arguments. Use the length field to determine the length of an array. Manipulate arrays with loops. Write methods to manipulate arrays. Create parallel arrays and two-dimensional arrays. Declare and initialize two-dimensional arrays.
---	---

Dates: February	Unit Plan: 6
------------------------	---------------------

Stage 1 – Desired Results

PA Standard(s)/Assessment Anchors Addressed:
15.4.12.H Use programming languages to develop logical thinking and problem solving skills.

15.4.8.J Create a complex computer program to solve a problem 3.4.12.C2 Apply the concept that engineering design is influenced by personal characteristics, such as creativity, resourcefulness, and the ability to visualize and think abstractly.	
Understanding(s): <i>Students will understand...</i> <ol style="list-style-type: none"> 1. The ability to simplify the writing of code is accomplished through inheritance and polymorphism. (3.4.12.C2) 2. Superclasses and subclasses is a relationship among classes. (15.4.8.J) 3. Using an abstract class can reduce the repetition of code throughout similar classes. (15.4.8.J) 4. Private fields in a superclass cannot be accessed in a subclass. (15.4.12.H) 	Essential Question(s): <ul style="list-style-type: none"> ▪ How can a program be made more flexible? ▪ When should information be shared among all instances of a class? ▪ How can the implementation of encapsulation and information hiding be used to reduce the amount of code? ▪ How is the distribution of responsibilities accomplished in Java?
Learning Objectives: <i>Students will know...</i> <ul style="list-style-type: none"> ▪ When it is appropriate to include class (static) variables and methods in a class. (15.4.12.H) ▪ The role of Java interfaces in a software system and define an interface for a set of implementing classes. (15.4.8.J) ▪ The use of inheritance by extending a class. (15.4.8.J) ▪ The use of polymorphism and know how to override methods in a superclass. (3.4.12.C2) ▪ How to place the common features (variables and methods) of a set of classes in an abstract class. (15.4.8.J) ▪ The implications of reference types for equality, copying, and mixed-mode operations. (15.4.8.J) 	Students will be able to: <ul style="list-style-type: none"> ▪ Know when it is appropriate to include class (static) variables and methods. ▪ Create classes that inherit another. ▪ Understand the role of Java interfaces in a software system and define an interface for a set of implementing classes. ▪ Refine a program with and abstract class by using subclasses. ▪ Understand the use of inheritance by extending a class. ▪ Understand the use of polymorphism and know how to override methods in a superclass. ▪ Use “super” to call public methods of a superclass from within a subclass. ▪ Access a superclass’s protected fields through a subclass. ▪ Place the common features (variables and methods) of a set of classes in an abstract class. ▪ Understand the implications of reference types for equality, copying, and mixed-mode operations. ▪ Distinguish between abstract, final, overriding, and shadowing as applied to classes and methods.
Dates: March	Unit Plan: 7
Stage 1 – Desired Results	
PA Standard(s)/Assessment Anchors Addressed: 15.4.12.H Use programming languages to develop logical thinking and problem solving skills. 15.4.8.J Create a complex computer program to solve a problem 3.4.12.C2 Apply the concept that engineering design is influenced by personal characteristics, such as creativity, resourcefulness, and the ability to visualize and think abstractly. 3.4.12.C3 Apply the concept that many technological problems require a multi-disciplinary approach.	

<p>Understanding(s): <i>Students will understand...</i></p> <ol style="list-style-type: none"> 1. Recursion can solve complex mathematical problems. (15.4.8.J) 2. The similarities and differences between recursive and iterative solutions of a problem. (3.4.12.C2) 3. Always check and test a recursive method for correctness. (3.4.12.C3) 4. How a computer executes a recursive method. (15.4.12.H) 5. The behavior of a complex sort a search algorithm. (15.4.8.J) 	<p>Essential Question(s):</p> <ul style="list-style-type: none"> ▪ Why is it important to use recursion in a program? ▪ How can a recursive method be used to solve complex mathematical problems? ▪ How can a large array be sorted efficiently? ▪ What are the benefits of using a binary search over other searching methods?
<p>Learning Objectives: <i>Students will know...</i></p> <ul style="list-style-type: none"> • How to write methods to perform insertions and removals at given positions in an array. (3.4.12.C2) • The issues involved when working with arrays of objects. (3.4.12.C3) • How to perform simple operations with Java's ArrayList class. (15.4.12.H) • How to use string methods appropriately. (3.4.12.C2) • A recursive method is a method that calls on itself to solve a problem. (15.4.8.J) • How to use a recursive method to solve a complex mathematical problem. (15.4.8.J) • How to implement quicksort and binary search into a program. (15.4.8.J) • Quicksort is a sort algorithm that uses recursion and can perform much more efficiently than any other sorting algorithm. (3.4.12.C2) 	<p>Students will be able to:</p> <ul style="list-style-type: none"> ▪ Construct programs that can search an array using binary searching. ▪ Construct programs that sorts an array using bubble, selection, and insertion sorting methods. ▪ Construct programs that can efficiently sort large arrays using quicksort. ▪ Construct a program that uses recursion. ▪ Construct a program that uses array lists to organize values.
<p>Dates: April/May</p>	<p>Unit Plan: 8</p>
<p>Stage 1 – Desired Results</p>	
<p>PA Standard(s)/Assessment Anchors Addressed: 15.4.12.H Use programming languages to develop logical thinking and problem solving skills. 15.4.8.J Create a complex computer program to solve a problem 3.4.12.C2 Apply the concept that engineering design is influenced by personal characteristics, such as creativity, resourcefulness, and the ability to visualize and think abstractly. 3.4.12.C3 Apply the concept that many technological problems require a multi-disciplinary approach.</p>	

<p>Understanding(s): Students will understand...</p> <ol style="list-style-type: none"> 1. Chatbots can be programmed to have a full conversation with an individual. (3.4.12.C3) 2. Using arrays can simplify the code for random responses. (3.4.12.C2) 3. How to modify a digital picture using java. (15.4.8.J) 4. How to use two dimensional arrays. (15.4.8.J) 5. How arraylists can be used to simplify code. (15.4.8.J) 6. Using inheritance and abstract classes can significantly reduce the length of code. (15.4.12.H) 	<p>Essential Question(s):</p> <ul style="list-style-type: none"> ▪ How can the MagPie chatbot code be altered to respond better to inputs? ▪ How can digital pictures be manipulated using java? ▪ How can the Elevens card game be programmed using Java?
<p>Learning Objectives: Students will know...</p> <ul style="list-style-type: none"> ▪ How to use string methods to allow the chatbot to respond to different statements. (3.4.12.C3) ▪ Arrays can store strings, which can be used for random responses. (3.4.12.C2) ▪ Digital pictures can be modified using java. (15.4.8.J) ▪ Pictures can be changed by altering one pixel at a time..(15.4.8.J) ▪ Two-dimensional arrays can hold a variety of information, such as pixels. (15.4.8.J) ▪ Math.random can be used to shuffle a deck of cards. (15.4.8.J) ▪ Abstract classes can simplify code by sharing methods. (15.4.12.H) ▪ ArrayLists can be used to store Objects. (15.4.8.J) 	<p>Students will be able to:</p> <ul style="list-style-type: none"> ▪ Write a program that creates a chatbot. ▪ Add multiple responses along with random responses to the chatbots vocabulary. ▪ Modify a digital picture using java. ▪ Modify the color of a picture. ▪ Use two-dimensional arrays to store data. ▪ Design and develop the Elevens card game. ▪ Create a deck of cards using an array list. ▪ Create a program that shuffles a deck of cards.