Davison Community Schools
ADVISORY CURRICULUM COUNCIL
Phase II, *date*

Course
Phase I: Course Essential Questions How can computers be used as a tool to help solve a variety of problems?

Phase II Curriculum

Unit: 1 - Introduction to Computer Programming

Essential Questions:	**Essential Understanding:**
1. What components are required for all computer programs? 2. What is the difference between low level programming languages and high-level languages. 3. What are the common design tools that are used to create a model of a program?	• Every program must contain two components, which are the "Using namespace std;" clause at the beginning of the program and a Main function with an integer return type. • A low-level language resembles the numeric machine language of the computer more than the natural language of humans. High-level languages are easier for humans to learn because they are closer to the level of human readability. • Programs can be modeled using the following design tools: hierarchy chart, flowchart, pseudocode, and outlines, which help make creating the program easier.

Curriculum Standards- DOK noted where applicable with Standards

Knowledge/Content **I Know** …(includes academic vocabulary)	**Skills/Processes** **I Can** …
• The two components needed for every program. - Using namespace std; - Int Main() • Examples of common low level languages. - C - Assembly • Examples of common high level languages. - Java - Visual Basic	• Write a simple program that outputs, "Hello World!" • Design program models using common design tools. • Recognize situations in their life where using a computer program would be helpful.

Course

Phase I: Course Essential Questions

How can computers be used as a tool to help solve a variety of problems?

Phase II Curriculum

Unit: 2 – Introduction to the C++ Language

Essential Questions:	**Essential Understanding:**
1. What are the five main data types? 2. What is a literal? 3. What is the main difference between the int & double data types? 4. What is the main difference between the char & string data types? 5. All char literals must be enclosed in what character? 6. All string literals must be enclosed in what character?	• The 5 main data types are: bool, int, double, char, string. • A literal is information that can be stored in a variable. • The main difference between the int and double data types is that int variables contain integers and double variables contain numbers containing decimals. • The main difference between the char and the string data types is that char variables contain a single character and string variables contain more than one character. • A char literal must be enclosed in single quotes whenever it is used in a program. • A string literal must be enclosed in double quotes whenever it is used in a program.

Curriculum Standards- DOK noted where applicable with Standards

Knowledge/Content **I Know** …(includes academic vocabulary)	**Skills/Processes** **I Can** …
The 5 Main Data Types in the C++ language: • Bool (holds values, “true” or “false”) • Char (holds single character values) • Int (holds integer values) • Double (holds decimal values) • String (holds multi-character values)	• Give examples of literals for each of the five data types. • Create and use variables for each of the five data types. • Use variables of correct data type to store a specific literal.

Course

Phase I: Course Essential Questions

How can computers be used as a tool to help solve a variety of problems?

Phase II Curriculum

Unit: 3 – Expression and Interactivity

Essential Questions:	**Essential Understanding:**
1. What must a program include in order to receive information from a user via the keyboard and/or display information on the computer screen? 2. What command is used to receive information from a user via the keyboard? 3. What command is used to display information to the computer screen? 4. What must a program include in order for it to have the ability to format user input as well as information displayed on the screen?	• A program must import the input/output library by writing "#include <iostream>" at the very top of its code to receive information from a user via the keyboard and/or to display information on the computer screen. • The "cin >>" command is used to receive information from a user via the keyboard. • The "cout <<" command is used to display information to the computer screen. • A program must import the formatting library by writing "#include <iomanip>" at the very top of its code in order to manipulate information that is being received from a user or sent to the computer screen

Curriculum Standards- DOK noted where applicable with Standards

Knowledge/Content **I Know** …(includes academic vocabulary)	**Skills/Processes** **I Can** …
• Commands from the "iostream" library - cout << - cin >> • Commands from the "iomanip" library - setw () - set precision () • Commands from the "cmath" library - pow () - sqrt () - sin, cos, tan	• Receive information from a user using the "cin >>" command. • Send information to the computer screen using the "cout <<" command. • Format user input and output using the commands from the "iomanip" library. • Create programs that solve math programs at the high school level (Algebra 1 and above) using the commands available in the "cmath" library.

Course

Phase I: Course Essential Questions

How can computers be used as a tool to help solve a variety of problems?

Phase II Curriculum

Unit: 4 – Selection Structures (Making Decisions)

Essential Questions:	**Essential Understanding:**
1. What programming component prevents a programming from running all code from top to bottom, but instead causes it to run certain code based on various situations? 2. How does a program differentiate between various situations during run-time in order to run the appropriate code in an If/Else IF/Else statement block and a switch statement? 3. How are variables and/or literals compared when executing a program?	• An If/Else IF/Else statement block can be used to cause a programming to run certain segments of code as well as exclude other segments of code based on various situations. • A switch statement can also be used to run desired segments of code as well as exclude segments of code that are not desired based on the value of integer or character variables. • A program uses logical operators and compares variables/literals using relational operators to determine the correct code to execute during run-time. • Variables and/or literals are compared using relational operators.

Curriculum Standards- DOK noted where applicable with Standards

Knowledge/Content **I Know** …(includes academic vocabulary)	**Skills/Processes** **I Can** …
• When to use "If" statements • Wen to use "If / Else" statements • When to use "If / Else If / Else" statement blocks • Logical Operators - AND (&&) - OR (||) - NOT (!) • Relational Operators - EQUAL TO (==) - NOT EQUAL TO (!=) Switch • Case statements • Break statements • Default statements	• Write programs containing If/Else IF/Else statement blocks • Writing programs containing switch statements. • Use logical operators to cause a program to run certain segments of code for a specific situation. • Compare variables and/or literals using relational operators. • Write programs containing switch statement blocks

Course

Phase I: Course Essential Questions

How can computers be used as a tool to help solve a variety of problems?

Phase II Curriculum

Unit: 5 – Repetition Structures (Looping)

Essential Questions:	**Essential Understanding:**
1. What programming component is used to run the same code for a set number of iterations or until a condition is met? 2. What are the 3 types of loops? 3. When should a For loop be used? 4. When should a While loop be used? 5. When should a Do-While loop be used? 6. What type of variable can be used to determine how many iterations a loop has performed? 7. What statement causes a loop to terminate prior to its primary condition becoming false?	• A loop is used to run a segment of code for a set number of iterations or until a condition is met. • The 3 types of loops are For, While, and Do-While. • A For loop should be used when the total number of iteration is known and the desire is for the first iteration to execute only if the primary condition is true. • A While loop should be used when the total number of iterations is NOT known and the desire is the first iteration to execute only if the primary condition is true. • The Do-While loop should be used when the total number of iterations is NOT known and the desire is for the first iteration to execute automatically without checking the primary condition. • An integer variable (often called a counter) can be placed inside a loop to determine how many iterations it has performed. • The break statement causes a loop to terminate early prior to its primary condition becoming false.

Curriculum Standards- DOK noted where applicable with Standards

Knowledge/Content **I Know** …(includes academic vocabulary)	**Skills/Processes** **I Can** …
The three types of loops: • For loop • While Loop • Do-While Loop Components associated with loops: • Counter variable (integer variable) • Break Statement	• Write programs that use each of the 3 types of loops • Use a counter variable to determine how many iterations a loop has completed. • Use a break statement to terminate a loop.

Course

Phase I: Course Essential Questions

How can computers be used as a tool to help solve a variety of problems?

Phase II Curriculum

Unit: 6 – Functions

Essential Questions:	**Essential Understanding:**
1. What does it mean to modularize a program? 2. What is a function and how is it executed? 3. What is the benefit to having a function or multiple functions in a program? 4. What components are necessary to create and execute a single function. 5. What are global and local variables and when are the used?	• A program is modularized if its processes are divided up into functions. • A function is a segment of code that does a specific task. It is executed using a function call. • The benefit of having functions in a program is that the code inside the function can be reused and executed repeatedly by simply using a function call. • Each function in a program must have the following components in order to run correctly: header, body, return type, and prototype. • Global variables are declared outside of functions and should only be used when containing a constant. Local variables are declared inside functions and are destroyed when the function ends.

Curriculum Standards- DOK noted where applicable with Standards

Knowledge/Content **I Know** …(includes academic vocabulary)	**Skills/Processes** **I Can** …
• The components that make up each function: - header, body, return type, prototype. • Implementation of a Function Call • Techniques for sending/receiving variables/literals from a function. - "Passing by Value" - "Passing by Reference" • Local, Global, & Static Variables • Function Overloading	• Create and execute a function • Send variables/literals to a function. • Return a variable/literal from a function. • Write a program that contains and uses local and global variables • Write a program that contains and uses multiple functions. • Create a function that has been overloaded.

Course

Phase I: Course Essential Questions

How can computers be used as a tool to help solve a variety of problems?

Phase II Curriculum

Unit: 7 – Arrays

Essential Questions:	**Essential Understanding:**
1. What is an array? 2. What information must be included when creating an array. 3. How is data in an array stored and accessed?	• An array is a single object that stores multiple values of the same data type. • When declaring an array, its data type and size must be known and included. • An array is divided into elements. Each unit of data is stored in a single element and accessed using the element's index value.

Curriculum Standards- DOK noted where applicable with Standards

Knowledge/Content **I Know** …(includes academic vocabulary)	**Skills/Processes** **I Can** …
• Declaration and Initialization of an Array • Common components associated with an array, such as: - Array Elements - Index Values • Declaration and Initialization of Multi-Dimensional Arrays	• Create an array of specified size and data type. • Store data in an array • Access data from an array • Use an array in a program to accomplish a specified task.

Course

Phase I: Course Essential Questions

How can computers be used as a tool to help solve a variety of problems?

Phase II Curriculum

Unit: 8 – Searching and Sorting Arrays

Essential Questions:	**Essential Understanding:**
1. What must a program include in order to receive information from a file and/or write information to a file? 2. What type of object is used to read information from a file? 3. What type of object is used to write information from a file? 4. What are the two sorting algorithms for sorting	• A program must import the file stream library by writing "#include <fstream>" at the very top of its code to receive information from a file and/or write information to a file. • An input file stream object (ifstream) uses the open() function to begin reading information from a file. • An output file stream object (ofstream) uses the

data in an array? 5. In what situation should the Bubble sort algorithm be used to sort data? 6. In what situations should the Selection sort algorithm be used to sort data? 7. What are the two searching algorithms that are used for searching for a target value in an array? 8. In what situation should the Linear search algorithm be used to search for a target value in an array? 9. In what situation should the Binary search algorithm be used to search for a target value in an array?	open() function to open an output file and begin writing information to it. • The Bubble sort and the Selection sort can be used to sort data contained in an array. • The Bubble sort algorithm should be used to sort data that is already partially sorted. • The Selection sort algorithm should be used to sort data that is NOT partially sorted. • The Linear search and the Binary search can be used to search for a target value in an array. • The Linear search algorithm should be used when the data is not sorted and/or the speed in which the data is searched is irrelevant. • The Binary search algorithm should be used when the data is sorted and the speed in which the data is search is important
Curriculum Standards- DOK noted where applicable with Standards	
Knowledge/Content **I Know** …(includes academic vocabulary)	**Skills/Processes** **I Can** …
The objects required to read/write to/from a file: • Ifstream object • Ofstream object Two popular algorithms for sorting an array: • Bubble sort • Selection sort Two popular algorithms for searching an array: • Linear search • Binary search • The definition of a “Target value”.	• Create a program that reads data from a file using an ifstream object, stores it into an array, sorts the array using the proper sort, and writes the data to an output file using an ofstream object. • Create a program that searches data stored in an array using the appropriate search to find a specified target value.
Course	
Phase I: Course Essential Questions How can computers be used as a tool to help solve a variety of problems?	
Phase II Curriculum **Unit: 9 – Strings**	

Essential Questions:	Essential Understanding:
1. What is a string variable? 2. How are string variables declared? 3. How are string variables related to arrays?	• That there is a distinction between a string literal and an array of characters.

Curriculum Standards- DOK noted where applicable with Standards

Knowledge/Content **I Know** …(includes academic vocabulary)	Skills/Processes **I Can** …
Recognize misuse of the assignment statement with character data. The functions to manipulate a string: assign, substr, erase, find, getline, ignore, insert, length, replace. The term concatenation. The concatenation operator. The term consuming the character.	• Declare string variables and named constants. • Get string input using the getline function. • Ignore characters using the ignore function. • Determine the number of characters in a string. • Access the characters in a string. • Search a string. • Remove characters from a string. • Convert a string to a numeric data type. • Replace character in a string. • Insert characters within a string. • Duplicate characters within a string. • Concatenate strings.

Course

Phase I: Course Essential Questions

How can computers be used as a tool to help solve a variety of problems?

Phase II Curriculum

Unit: 10 – Structures and Classes

Essential Questions:	Essential Understanding:
1. What is a structure? 2. How are structures declared? 3. How are individual members initialized and accessed in a structure? 4. What are some of the common ways structures are used in a program? 5. What is an abstract data type (classes)? 6. What is OOPs programming? 7. What is an object? 8. How do you access the private data variables within your classes? (methods)	• A structure is an abstract data type that allows several variables with different data types, known as members, to be grouped into one single object. • Structures are defined at the top of a program's code and then can be declared in any function by using the structure's name followed by its identifier. • Members in a structure are initialized and accessed using the structure's identifier and dot operator (a period). This process allows data to be assigned to members in a structure and retrieved from members in a structure.

	• Structures can be passed to functions, returned from functions, stored in an element of an array, and compared to other structures using relational operators.
Curriculum Standards- DOK noted where applicable with Standards	
Knowledge/Content **I Know** …(includes academic vocabulary)	**Skills/Processes** **I Can** …
• Create and initialize a Structure • Store/Access literals from a Structure's Members • Dot operator is used to store/access literals from the members of a structure. • Structure identifier is used to declare a structure and used in the process to store/access literals from the members of a structure.	• Define a structure • Create a structure and store/access information in/from its members using the structure's identifier and dot operator. • Send/return a structure to/from a function. • Store a structure in an array. • Write a program that uses structures to tally how often each number in a list occurs. • Define a structure
Course	

**An estimated budget needs to be submitted along with a Phase II report.