CHINO VALLEY UNIFIED SCHOOL DISTRICT
INSTRUCTIONAL GUIDELINES
COMPUTER PROGRAMMING BASICS

Course Number	5E02
Department	Technology
Prerequisite	C or better in Algebra I
Length of Course	Two (2) semesters/One (1) year
Grade Level	10-12
Credit	5 units per semester/10 total units – elective
Repeatable	Not repeatable for credit
UC/CSU	Meets the "g" elective requirement
Board Approved	May 24, 2012

Description of Course – Computer Programming Basics will introduce computer-programming basics in Visual Basic language, as well as exposure to C++, Java, and HTML/Javascript. This course will provide a foundation in the programming language and expose students to computer-programming in other languages.

Rationale for Course – As our economy becomes more digitally-based, information-driven, and computer-dependent, the success of our nation in a global economy will depend on being able to adapt and innovate in the technology age. It is imperative that students be able to program the computer languages of today to create the innovations of tomorrow. Those that can successfully adapt will ensure that our nation, state, and region remain globally competitive and technologically innovative.

Standard 1 – Understand the strategies necessary to define and analyze systems and software requirements.

1.1 Objective: Know what a computer is, and the process of writing and executing a computer program.

1.1.1 Performance Indicator: Student will know the basic data processing cycle – input, process, and output.

1.1.2 Performance Indicator: Student will know the function of a computer program.

1.1.3 Performance Indicator: Student will understand the operational capabilities of a computer system – input/output, arithmetic, and logical operations.

1.1.4 Performance Indicator: Student will understand SAVE, LOAD, EDIT, KILL, and DIRECTORY.

1.2 Objective: Understand the elements of the BASIC language required to implement the logic of computer programming.

1.2.1 Performance Indicator: Student will understand program development and coding process.

1.2.2 Performance Indicator: Student will understand the logic required to create a list of information using a simple loop.

1.2.3 Performance Indicator: Student will understand and use the following BASIC statements: REM, DTA, READ, IF, PRINT, GO TO, and END.

1.2.4 Performance Indicator: Student will understand and use numeric and string variables and contents.

1.2.5 Performance Indicator: Student will understand and recognize good programming techniques, including program comments and proper indentation of the source code.

Standard 2 – Understand the creation and design of a software program.

2.1 Objective: Understand how to perform arithmetic operations, semicolon and tab functions.

2.1.1 Performance Indicator: Student will perform arithmetic operations.

2.1.2 Performance Indicator: Student will understand how to round numeric values.

2.1.3 Performance Indicator: Student will use the PRINT USING statements for editing fields.

2.1.4 Performance Indicator: Student will use the TAB statement for controlling output.

2.1.5 Performance Indicator: Student will design a program requiring calculations, accumulations, and printing final totals.

2.2 Objective: Compare values and perform alternative operations based upon the results of the comparison.

2.2.1 Performance Indicator: Student will understand the IF-THEN-ELSE logic structure which includes nesting.

2.2.2 Performance Indicator: Student will know and use the single entry/single exit rule for the IF-THEN-ELSE logic structure.

2.2.3 Performance Indicator: Student will use the BASIC IF statement and know how it should be written.

2.2.4 Performance Indicator: Student will understand relational operators.

2.2.5 Performance Indicator: Student will understand the difference between string comparisons, and numeric comparisons.

2.2.6 Performance Indicator: Student will understand the logical operators AND, OR, and NOT.

2.2.7 Performance Indicator: Student will know the manner of internally storing numeric data and considerations when comparing this data.

2.3 Objective: Understand the proper structure of loops.

2.3.1 Performance Indicator: Student will understand the interactive process.

2.3.2 Performance Indicator: Student will use the input statement with appropriate prompting.

2.3.3 Performance Indicator: Student will understand loops and the loop logic structure.

2.3.4 Performance Indicator: Student will use the “FOR” and “NEXT” statements for loops.

2.4 Objective: Understand and use arrays.

2.4.1 Performance Indicator: Student will define and load arrays.

2.4.2 Performance Indicator: Student will perform a search sequentially of an array for a known value and to extract the corresponding element from another array.

2.4.3 Performance Indicator: Student will understand binary search techniques.

2.4.4 Performance Indicator: Student will define and load multi-dimension arrays.

2.5 Objective: Use menus, subroutines, and sorting techniques.

2.5.1 Performance Indicator: Student will use menus in interactive programming.

2.5.2 Performance Indicator: Student will design and code programs using the case structure.

2.5.3 Performance Indicator: Student will use subroutines.

2.5.4 Performance Indicator: Student will understand sorts and design an exchange sort.

2.5.5 Performance Indicator: Student will design a program by decomposing the program into a series of functional modules.

2.6 Objective: Understand and use strings.

2.6.1 Performance Indicator: Student will create a personalized letter using the string functions.

2.6.2 Performance Indicator: Student will edit input data using the string functions.

2.6.3 Performance Indicator: Student will use string functions available with BASIC interpreters.

2.6.4 Performance Indicator: Student will search strings for delimiters and substrings.

2.6.5 Performance Indicator: Student will understand the design process for a program requiring numerous modules.

2.7 Objective: Use files, report generation, and functions in designing programs.

2.7.1 Performance Indicator: Student will use trig functions and other functions of the BASIC language that can be useful for certain applications.

2.7.2 Performance Indicator: Student will understand a control break process and the logic required to produce a control break report.

2.8 Objective: Create graphic designs.

2.8.1 Performance Indicator: Student will understand and use character strings.

2.8.2 Performance Indicator: Student will use read data commands.

2.8.3 Performance Indicator: Student will use SET-RESET and POKE-PEEK commands.

Standard 3 – Students will understand programming languages

3.1 Objective: Understand programming languages through exposure to: C++, HTML and Javascript, and Java.

3.1.1 Performance Indicator: Student will define C++, HTML and Javascript, and Java.