Arkansas CTE

Programming I: Grades 9, 10, 11, 12

Adopted 2007

Introduction to Programming

1.1 Define terminology

1. Prepare a list of terms with definitions 1.1.1

1.2 Explain the difference between system and application software

1. Identify various software as system or application 1.2.1

1.3 Discuss hardware and software

1. Identify technology as either hardware or software 1.3.1

1.4 Describe executable code and source code

1. Explain the difference between executable code and source code 1.4.1

1.5 Discuss Windows applications, events, and event driven programs

1. Give examples of events and the actions that result in an event-driven program 1.5.1

Programming Techniques and Characteristics of Good Programs

2.1 Define terminology

1. Prepare a list of terms with definitions 2.1.1

2.2 List the steps of the programming process

1. When given an example, be able to identify the correct steps 2.2.1

2.3 Identify syntax of comments

1. Use appropriate syntax to include comments in programs 2.3.1

2.4 Explain the characteristics of user-friendly programs

1. Write programs that have clear instructions 2.4.1
2. Write programs whose output is easy to read and understand 2.4.2

2.5 Explain the importance of program documentation and maintenance

1. Write programs that contain comments 2.5.1
2. Update an existing program 2.5.2

2.6 Explain the importance of algorithm and/or pseudocode in program development

1. Write a psuedocode (algorithm) for a programming problem 2.6.1

2.7 Identify different types of errors (syntax, semantic, run-time, compile time)

1. When given an example, identify the error type 2.7.1

2.8 Explain the characteristics of readable programs

1. Explain the characteristics of readable programs 2.8.1
2. Declare and use variables 2.8.2
3. Document difficult logic to make it easy to follow 2.8.3
4. Use standard naming conventions for controls by beginning the name with the appropriate prefix (btn for button, lbl for label, etc.) 2.8.4
5. Use the line continuation character (_) in the code to make it more readable 2.8.5

Data Types, Variables, Constants, and Mathematical Operations

3.1 Define terminology

1. Prepare a list of terms with definitions 3.1.1

3.2 List the following data types: Integer, Boolean, Single/Double, String, Char

1. Compare the data types 3.2.1
2. Determine whether a particular "number" would be considered numeric 3.2.2
3. Determine whether a number should be treated as an integer or a floating point (i.e. single, double) 3.2.3
4. Designate data type using correct syntax 3.2.4
5. Determine whether an identification number (such as Social Security Number) should be treated as a string or number 3.2.5
6. Explain the similarities and differences between Strings and Chars 3.2.6

3.3 Describe how to declare a variable (using Dim) and list the initial value for each type of data

1. Write programs in which variables are declared 3.3.1

3.4 Describe the syntax of an assignment statement

1. Write a program that uses assignment statements 3.4.1

3.5 Describe concatenation and list the concatenation operator (&)

1. Write output lines that use concatenation 3.5.1

3.6 Explain the advantages of using integer variables whenever possible (faster computation, require less memory, obtain exact answers)

1. Use integer variables in programs where appropriate `3.6.1`

3.7 Explain the advantages and disadvantages of floating-point numbers (round-off errors, more memory, approximate answers, slower computation, size of numbers to be stored, etc.)

1. Use floating point variables in programs where appropriate `3.7.1`

3.8 List arithmetic operations and order of operations (^, *, /, \, mod, +, -)

1. Write formulas using operators and order of operations `3.8.1`
2. Write programs that use mathematical operations correctly (integer arithmetic vs floating point arithmetic) `3.8.2`

3.9 Describe the syntax for assignment statement

1. Write statements that assign values to objects, such as text to a label `3.9.1`
2. Write statements that assign values to variables `3.9.2`

3.10 Explain rules for choosing variable names

1. Write programs that use proper variable naming conventions `3.10.1`
2. Write programs that use descriptive variable names `3.10.2`

3.11 Explain the circumstances and give examples of appropriate occasions to use constants

1. Use appropriate constants when writing programs `3.11.1`

3.12 List and explain the function of the following constants: vbCrLF, vbTab, Nothing

1. Write programs that use these constants `3.12.1`

3.13 Display the results of calculations formatted appropriately (Currency, Standard, Percent)

1. Write programs where the results of calculations are formatted in Currency, Standard, and Percent formats `3.13.1`

3.14 Describe the circumstances under which variables should be declared locally and globally

1. Write programs where the variables are declared appropriately (either globally at the beginning of the program or locally within the procedure) `3.14.1`

3.15 Explain automatic type conversion (for example, an integer assigned to double variable, or a double assigned to an integer variable)

1. Determine the value stored in an integer variable when a double is used `13.15.1`
2. Determine the value stored in a double variable when an integer is used `13.15.2`

3.16 Explain the purpose of the val() function and explain what is stored when non-numeric data is used

1. Write programs that use the val() function to cast text to a number `13.16.1`

3.17 Explain the use of the IsNumeric function

1. Write programs that use the IsNumeric() function to check the data prior to use in a mathematical formula or numeric variable `3.17.1`

3.18 Explain the uses for random numbers and the purpose of the Randomize() statement

1. Write programs that use random numbers `3.18.1`

3.19 Explain how Int() and Fix() functions work

1. Predict the results of using Fix() and Int() with a group of real (floating point) numbers `3.19.1`

3.20 Explain the scope and lifetime of static variables

1. Write programs that use static variables `3.20.1`

3.21 Explain the purpose of a counter variable

1. Write programs that use static variables as counters `3.21.1`

3.22 Explain the purpose of an accumulator variable

1. Write programs that use static variables as accumulators `3.22.1`

Simple Programs and Visual Basic Features

4.1 Define terminology

1. Prepare a list of terms with definitions `4.1.1`

4.2 Describe the process of creating a new project

1. Create a new Visual Basic project `4.2.1`

4.3 Describe the process of creating/designing a form

1. Create or design a form `4.3.1`
2. Place and size controls on the form `4.3.2`
3. Set properties in the properties window `4.3.3`

4.4 Explain the difference in setting properties at design time and setting/changing properties at runtime

1. Create a program where properties are set both in the design phase and set or modified in the code `4.4.1`

4.5 Describe the difference in the name and text property

1. Name controls using the convention of beginning the object name with a prefix, such as btn for button or mnu for menu `4.5.1`
2. Use the text property to change the text on various object, both setting it at design time and in the code `4.5.2`

4.6 Describe some actions that can trigger event procedures

2. Write click event procedures `4.6.2`

4.7 Explain how to use an assignment statement to change an object's value at runtime, including the use of Me, dot (.) notation, the equal (=) sign, and IntelliSense

1. Write program statements that change the values of properties at runtime, using Me, dot notation, equal sign, and IntelliSense `4.7.1`

4.8 Explain the use and features of MainMenu

1. Write programs that contain a MainMenu `4.8.1`

4.9 List several properties that can be used to improve the appearance of objects (Font and font features, BackColor, ForeColor, TextAlign)

1. Use these features in programs to improve the appearance of the forms and the objects on the form `4.9.1`

4.10 Describe the purpose of a label

1. Write programs that use labels as prompts `4.10.1`
2. Write programs that use labels to display answers `4.10.2`

4.11 Explain the use of a button

1. Write programs that use buttons and code the click event procedure `4.11.1`

4.12 Explain the uses for a TextBox

1. Write programs that use text boxes for user input `4.12.1`

More Visual Basic Features

5.1 Define terminology

1. Prepare a list of terms with definitions `5.1.1`

5.2 Explain the use of the checked property--for example with a RadioButton or CheckBox

1. Use the checked property in code to determine which object is selected `5.2.1`

5.3 Describe the features of a RadioButtons and a GroupBox

1. Write programs that use RadioButtons--code both the click event and use the checked property to determine the item selected 5.3.1
2. Use a GroupBox for a set of RadioButtons 5.3.2

5.4 Describe a MessageBox

1. Write code that shows a message box to display information or a warning, setting the text, caption, and icon 5.4.1

5.5 Describe the use for a PictureBox

1. Design a form with a PictureBox with a graphic image 5.5.1
2. Place the image in the bin folder 5.5.2
3. Set the SizeMode to Normal, StretchImage, AutoSize, or CenterImage 5.5.3

5.6 Describe the purpose for an InputBox

1. Write programs that use an InputBox 5.6.1

5.7 Describe the type of data returned by an InputBox and the needed steps to convert to the String data to numeric data

1. Write programs where the data being returned is being used as a String 5.7.1
2. Write programs where the expected data is numeric, using IsNumeric() to check validity, and val() to convert to a number 5.7.2

Decision Structure

6.1 Define terminology

1. Prepare a list of terms with definitions 6.1.1

6.2 List relational operators

1. Write Boolean expressions that use the appropriate relational operator 6.2.1

6.3 Describe the process of comparing two strings

1. When given two strings, determine if they are equal, the first is smaller, or the first is larger 6.3.1

6.4 Describe a roundoff error

1. Write statements that avoid the problems if created by roundoff errors 6.4.1

6.5 Explain the syntax and logic of if statements

1. Write programs that use statements if 6.5.1

6.6 Explain the syntax and logic of if-else statements

1. Write statements that use if-else to make the correct decision based on the data 6.6.1

6.7 Explain the use of logical operators and, or, and not

1. Write programs which require the use and, not, and or `6.7.1`

6.8 Explain the use of the select case statement

1. Write programs that use the select case statement, including cases that use a range of values, a list of possible values, and case else `6.8.1`

Loops

7.1 Define terminology

1. Prepare a list of terms with definitions `7.1.1`

7.2 Describe the purpose and syntax of for-next loops

1. Write programs that use for-next loops `7.2.1`

7.3 Explain the procedure to use for loops to count in increments/decrements other than one

1. Write counting for loops with increments other than 1 `7.3.1`

7.4 List the types of do loops

1. Write programs that use do loops `7.4.1`

7.5 Describe the use of a do until loop

1. Write programs that use do…loop until `7.5.1`
2. Write programs that use do until…loop `7.5.2`

7.6 Describe the difference in a do until and a do while loop

1. Write programs that use do…loop while `7.6.1`
2. Write programs that use do while…loop `7.6.2`

7.7 Describe the difference in a entrance condition loop and an exit condition loop

1. Explain the potential effects of placing the condition on the do line and the loop line and conditions where each is appropriate `7.7.1`

7.8 Describe a loop which ends when a sentinel or flag is entered

1. Write loops that end when a sentinel or flag is entered `7.8.1`
2. Write loops that use an InputBox and end when the Cancel button is selected `7.8.2`

7.9 Explain the process of using counters with loops

1. Write programs that use counters with loops `7.9.1`

7.10 Explain the logic of using accumulators with loops

1. Write programs that use accumulators with loops `7.10.1`

7.11 Explain the difference in the effect of a while loop (entrance condition loop) and a do while (exit condition loop) loop

1. Write programs that use do while loops `7.11.1`

7.12 Explain the syntax of nested loops

1. Determine the output of a nested loop `7.12.1`
2. Write programs that use nested for loops `7.12.2`