PROGRAMMING I – VISUAL BASIC

Curriculum Content Frameworks

Please note: All assessment questions will be taken from the knowledge portion of these frameworks.

Prepared by

Marilyn Carrell, Springdale High School
Carl Frank, Arkansas School for Mathematics, Sciences, and the Arts
Kimberly Raup, Conway High School West

Facilitated by

Karen Chisholm, Program Manager
Office of Assessment and Curriculum
Arkansas Department of Workforce Education

Edited by

Sandra Porter, Program Manager
Jim Brock, Program Advisor
Ginger Fisher, Program Advisor
LaTrenda Jackson, Program Advisor
Tim Johnston, Program Advisor
Office of Business and Marketing Technology
Arkansas Department of Workforce Education

Disseminated by

Career and Technical Education
Office of Assessment and Curriculum
Arkansas Department of Workforce Education

Curriculum Content Frameworks

PROGRAMMING I - VISUAL BASIC

Grade Levels: 9, 10, 11, 12 Course Code: 492390	Prerequisite: Geometry or Algebra II
Course Description: Programming I is a one-semester course in any modern, high-level, structured language. Concepts should be taught in the context of practical applications.	
An introduction to programming and problem solving. The language used is Visual Basic. No prior programming experience is required. Good programming style is stressed in a Windows environment. Topics included are: documentation of programs, structuring programs, program flow, decision structures, loops, and the more commonly used control objects. The contents of these frameworks are not intended to be taught in this order as independent units. Many of the skills are best introduced in one unit and then spiraled back to in future units with more complexity added. However, by the end of the semester all skills should be taught. This course was written for Visual Basic 2005 or later version. (Visual Basic 2005 Express can be downloaded from the Microsoft site.)	

Table of Contents

Unit 1: Introduction to Programming
Hours: 3

Terminology: Application software, Compiler, Event-driven, Events, Executable code, Hardware, High-level language, Low-level language, Object-Oriented Programming (OOP), Operating system, Software, Source code, System software, Unicode, Windows application

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do				ACADEMIC and WORKPLACE SKILLS What the Instruction Should Reinforce		
Knowledge		**Application**		**Skill Group**	**Skill**	**Description**
1.1	Define terminology	1.1.1	Prepare a list of terms with definitions	Foundation	Reading	Applies/Understands technical words that pertain to programming [1.3.6]
					Writing	Uses words appropriately [1.6.21]
1.2	Explain the difference between system and application software	1.2.1	Identify various software as system or application	Foundation	Reading	Analyzes and applies what has been read to specific task [1.3.2]
				Thinking Skills	Decision Making	Comprehends ideas and concepts related to software [4.2.2]
1.3	Discuss hardware and software	1.3.1	Identify technology as either hardware or software	Foundation	Reading	Applies/Understands technical words that pertain to hardware and software [1.3.6]
				Thinking	Knowing How to Learn	Applies new knowledge and skills to hardware and software [4.3.1]
1.4	Describe executable code and source code	1.4.1	Explain the difference between executable code and source code	Foundation	Reading	Applies/Understands technical words that pertain to executable code and source code [1.3.6]
				Thinking	Reasoning	Sees relationship between two or more ideas, objects, or situations [4.5.5]
1.5	Discuss Windows applications, events, and event driven programs	1.5.1	Give examples of events and the actions that result in an event-driven program	Foundation	Reading	Applies/Understands technical words that pertain to events and event-driven programs [1.3.6]
				Thinking	Reasoning	Sees relationship between two or more ideas, objects, or situations [4.5.5]

Unit 2: Programming Techniques and Characteristics of Good Programs
Hours: 4

Terminology: Algorithm, Code, Documentation, Line continuation character, Logic errors, Naming conventions, Program maintenance, Pseudocode, Run-time error, Statement, Syntax, Syntax errors, User-friendly

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do		**ACADEMIC and WORKPLACE SKILLS** What the Instruction Should Reinforce		
Knowledge	**Application**	**Skill Group**	**Skill**	**Description**
2.1 Define terminology	2.1.1 Prepare a list of terms with definitions	Foundation	Reading	Applies/Understands technical words that pertain to programming techniques [1.3.6]
			Writing	Uses words appropriately [1.6.21]
2.2 List the steps of the programming process	2.2.1 When given an example, be able to identify the correct steps	Foundation	Science	Solves practical problems using scientific methods and techniques [1.4.22]
			Writing	Organizes information into an appropriate format [1.6.10]
		Thinking	Reasoning	Sees relationship between two or more ideas, objects, or situations [4.5.5]
2.3 Identify syntax of comments	2.3.1 Use appropriate syntax to include comments in programs	Foundation	Writing	Communicates thoughts, ideas, or facts in written form in a clear, concise manner [1.6.6]
		Thinking	Knowing How to Learn	Applies new knowledge and skills to syntax of comments [4.3.1]
2.4 Explain the characteristics of user-friendly programs	2.4.1 Write programs that have clear instructions	Foundation	Writing	Communicates thoughts, ideas, or facts in written form in a clear, concise manner [1.6.6]
	2.4.2 Write programs whose output is easy to read and understand	Thinking	Knowing How to Learn	Applies new knowledge and skills to user-friendly programs [4.3.1]
2.5 Explain the importance of program documentation and maintenance	2.5.1 Write programs that contain comments	Foundation	Writing	Communicates thoughts, ideas, or facts in written form in a clear, concise manner [1.6.6]
	2.5.2 Update an existing program	Personal Management	Organizational Effectiveness	Comprehends the organization's modes of operation [3.3.5]
		Thinking	Problem Solving	Devises and implements a plan of action to resolve problem [4.4.3]

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do				ACADEMIC and WORKPLACE SKILLS What the Instruction Should Reinforce		
Knowledge		**Application**		**Skill Group**	**Skill**	**Description**
2.6	Explain the importance of algorithm and/or pseudocode in program development	2.6.1	Write a psuedocode (algorithm) for a programming problem	Thinking	Problem Solving	Applies new knowledge and skills to algorithms and pseudocodes [4.3.1]
2.7	Identify different types of errors (syntax, semantic, run-time, compile time)	2.7.1	When given an example, identify the error type	Thinking	Reasoning	Uses logic to draw conclusions from available information [4.5.6]
2.8	Explain the characteristics of readable programs	2.8.1	Explain the characteristics of readable programs	Foundation	Writing	Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
		2.8.2	Declare and use variables	Thinking	Knowing How to Learn	Applies new knowledge and skills to readable programs [4.3.1]
		2.8.3	Document difficult logic to make it easy to follow			
		2.8.4	Use standard naming conventions for controls by beginning the name with the appropriate prefix (btn for button, lbl for label, etc.)			
		2.8.5	Use the line continuation character (_) in the code to make it more readable			

Unit 3: Data Types, Variables, Constants, and Mathematical Operations
Hours: 13

Terminology: Accumulator, Boolean, Character, Concatenation, Constants, Counter, Data type, Dim, Fix(), Floating point (real), Global declaration, Int(), Integer, Integer division, IsNumeric() function, Local declaration, Mathematical operators, Modulus, Order of operations, Random numbers, Randomize(), Round-off errors, Static variables, String, Val() function, Variable

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do		**ACADEMIC and WORKPLACE SKILLS** What the Instruction Should Reinforce		
Knowledge	**Application**	**Skill Group**	**Skill**	**Description**
3.1 Define terminology	3.1.1 Prepare a list of terms with definitions	Foundation	Reading	Applies/Understands technical words that pertain to data types, variables, constants, and mathematical operations [1.3.6]
			Writing	Uses words appropriately [1.6.21]
3.2 List the following data types: Integer, Boolean, Single/ Double, String, Char	3.2.1 Compare the data types	Foundation	Arithmetic/ Mathematics	Comprehends a mathematical ideas and concepts related to data types [1.1.13]
	3.2.2 Determine whether a particular "number" would be considered numeric		Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]
	3.2.3 Determine whether a number should be treated as an integer or a floating point (i.e. single, double)	Thinking	Reasoning	Comprehends ideas and concepts related to data types [4.5.2]
	3.2.4 Designate data type using correct syntax			
	3.2.5 Determine whether an identification number (such as Social Security Number) should be treated as a string or number			
	3.2.6 Explain the similarities and differences between Strings and Chars			
3.3 Describe how to declare a variable (using Dim) and list the intital value for each type of data	3.3.1 Write programs in which variables are declared	Foundation	Writing	Communicates thoughts, ideas, or facts in written form in a clear, concise manner [1.6.6]
		Thinking	Decision Making	Comprehends ideas and concepts related to variable data [4.2.2]
3.4 Describe the syntax of an assignment statement	3.4.1 Write a program that uses assignment statements	Foundation	Writing	Communicates thoughts, ideas, or facts in written form in a clear, concise manner [1.6.6]
		Thinking	Decision Making	Comprehends ideas and concepts related to assignment statements [4.2.2]

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do				ACADEMIC and WORKPLACE SKILLS What the Instruction Should Reinforce		
Knowledge		**Application**		**Skill Group**	**Skill**	**Description**
3.5	Describe concatenation and list the concatenation operator (&)	3.5.1	Write output lines that use concatenation	Foundation	Writing	Communicates thoughts, ideas, or facts in written form in a clear, concise manner [1.6.6]
				Thinking	Decision Making	Comprehends ideas and concepts related to concatenation [4.2.2]
3.6	Explain the advantages of using integer variables whenever possible (faster computation, require less memory, obtain exact answers)	3.6.1	Use integer variables in programs where appropriate	Foundation	Arithmetic/ Mathematics	Comprehends a mathematical ideas and concepts related to integer variables [1.1.13]
					Writing	Uses words appropriately [1.6.21]
				Thinking	Decision Making	Comprehends ideas and concepts related to integer variables [4.2.2]
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.)	3.7.1	Use floating point variables in programs where appropriate	Foundation	Arithmetic/ Mathematics	Comprehends a mathematical ideas and concepts related to point variables [1.1.13]
					Writing	Uses words appropriately [1.6.21]
				Thinking	Decision Making	Comprehends ideas and concepts related to point variables [4.2.2]
3.8	List arithmetic operations and order of operations (^, *, /, \, mod, +, -)	3.8.1	Write formulas using operators and order of operations	Foundation	Arithmetic/ Mathematics	Comprehends a mathematical ideas and concepts related to arithmetic operations [1.1.13]
		3.8.2	Write programs that use mathematical operations correctly (integer arithmetic vs floating point arithmetic)	Thinking	Reasoning	Comprehends ideas and concepts related to arithmetic operations [4.5.2]
3.9	Describe the syntax for assignment statement	3.9.1	Write statements that assign values to objects, such as text to a label	Foundation	Arithmetic/ Mathematics	Comprehends a mathematical ideas and concepts related to syntax [1.1.13]
		3.9.2	Write statements that assign values to variables	Thinking	Reasoning	Comprehends ideas and concepts related to syntax [4.5.2]
3.10	Explain rules for choosing variable names	3.10.1	Write programs that use proper variable naming conventions	Foundation	Writing	Analyzes data, summarizes results, and makes conclusions [1.6.2]
		3.10.2	Write programs that use descriptive variable names	Thinking	Reasoning	Comprehends ideas and concepts related to variable names [4.5.2]

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do		ACADEMIC and WORKPLACE SKILLS What the Instruction Should Reinforce		
Knowledge	**Application**	**Skill Group**	**Skill**	**Description**
3.11 Explain the circumstances and give examples of appropriate occasions to use constants	3.11.1 Use appropriate constants when writing programs	Foundation Thinking	Arithmetic/ Mathematics Writing Decision Making	Comprehends a mathematical ideas and concepts related to constants [1.1.13] Uses words appropriately [1.6.21] Comprehends ideas and concepts related to constants [4.2.2]
3.12 List and explain the function of the following constants: vbCrLF, vbTab, Nothing	3.12.1 Write programs that use these constants	Foundation Thinking	Arithmetic/ Mathematics Reasoning	Comprehends a mathematical ideas and concepts related to constants [1.1.13] Comprehends ideas and concepts related to constants [4.5.2]
3.13 Display the results of calculations formatted appropriately (Currency, Standard, Percent)	3.13.1 Write programs where the results of calculations are formatted in Currency, Standard, and Percent formats	Foundation Thinking	Arithmetic/ Mathematics Reasoning	Comprehends a mathematical ideas and concepts related to calculations [1.1.13] Comprehends ideas and concepts related to calculations [4.5.2]
3.14 Describe the circumstances under which variables should be declared locally and globally	3.14.1 Write programs where the variables are declared appropriately (either globally at the beginning of the program or locally within the procedure)	Foundation Thinking	Arithmetic/ Mathematics Writing Decision Making	Comprehends a mathematical ideas and concepts related to variables [1.1.13] Uses words appropriately [1.6.21] Comprehends ideas and concepts related to variables [4.2.2]
3.15 Explain automatic type conversion (for example, an integer assigned to double variable, or a double assigned to an integer variable)	13.15.1 Determine the value stored in an integer variable when a double is used 13.15.2 Determine the value stored in a double variable when an integer is used	Foundation Thinking	Writing Reasoning	Presents answers/conclusions in a clear and understandable form [1.6.13] Comprehends ideas and concepts related to automatic type conversion [4.5.2]
3.16 Explain the purpose of the val() function and explain what is stored when non-numeric data is used	13.16.1 Write programs that use the val() function to cast text to a number	Foundation Thinking	Writing Reasoning	Uses words appropriately [1.6.2] Comprehends ideas and concepts related to functions [4.5.2]
3.17 Explain the use of the IsNumeric function	3.17.1 Write programs that use the IsNumeric() function to check the data prior to use in a mathematical formula or numeric variable	Foundation Thinking	Writing Reasoning	Presents answers/conclusions in a clear and understandable form [1.6.13] Comprehends ideas and concepts related to the IsNumeric function [4.5.2]

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do		ACADEMIC and WORKPLACE SKILLS What the Instruction Should Reinforce		
Knowledge	**Application**	**Skill Group**	**Skill**	**Description**
3.18 Explain the uses for random numbers and the purpose of the Randomize() statement	3.18.1 Write programs that use random numbers	Foundation	Arithmetic/ Mathematics	Comprehends mathematical ideas and concepts related to random numbers [1.1.13]
		Thinking	Reasoning	Comprehends ideas and concepts related to random numbers [4.5.2]
3.19 Explain how Int() and Fix() functions work	3.19.1 Predict the results of using Fix() and Int() with a group of real (floating point) numbers	Foundation	Arithmetic/ Mathematics	Comprehends mathematical ideas and concepts related to Int() and Fix() functions [1.1.13]
			Writing	Uses words appropriately [1.6.21]
		Thinking	Decision Making	Comprehends ideas and concepts related to Int() and Fix() functions [4.2.2]
3.20 Explain the scope and lifetime of static variables	3.20.1 Write programs that use static variables	Foundation	Writing	Uses words appropriately [1.6.2]
		Thinking	Reasoning	Comprehends ideas and concepts related to static variables [4.5.2]
3.21 Explain the purpose of a counter variable	3.21.1 Write programs that use static variables as counters	Foundation	Writing	Uses words appropriately [1.6.2]
		Thinking	Reasoning	Comprehends ideas and concepts related to counter variables [4.5.2]
3.22 Explain the purpose of an accumulator variable	3.22.1 Write programs that use static variables as accumulators	Foundation	Writing	Uses words appropriately [1.6.2]
		Thinking	Reasoning	Comprehends ideas and concepts related to accumulator variables [4.5.2]

Unit 4: Simple Programs and Visual Basic Features
Hours: 8

Terminology: BackColor, Button, Checked, Control, Design view, Event procedure, Font, ForeColor, Form, IntelliSense, Label, MainMenu, Me, Name, Project, Property, Text, TextAlign, TextBox, Toolbar, Toolbox

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do				**ACADEMIC and WORKPLACE SKILLS** What the Instruction Should Reinforce		
Knowledge		**Application**		**Skill Group**	**Skill**	**Description**
4.1	Define terminology	4.1.1	Prepare a list of terms with definitions	Foundation	Reading	Applies/Understands technical words that pertain to simple programs and visual basic features [1.3.6]
					Writing	Uses words appropriately [1.6.21]
4.2	Describe the process of creating a new project	4.2.1	Create a new Visual Basic project	Foundation	Writing	Organizes information into an appropriate format [1.6.10]
4.3	Describe the process of creating/designing a form	4.3.1	Create or design a form	Foundation	Writing	Composes and creates documents – letters, manuals, reports, proposals, graphs, flow charts, etc. [1.6.8]
		4.3.2	Place and size controls on the form			
		4.3.3	Set properties in the properties window	Thinking	Reasoning	Sees relationship between two or more ideas, objects, or situations [4.5.5]
4.4	Explain the difference in setting properties at design time and setting/changing properties at runtime	4.4.1	Create a program where properties are set both in the design phase and set or modified in the code	Foundation	Writing	Organizes information into an appropriate format [1.6.10]
				Thinking	Problem Solving	Devises and implements a plan of action to resolve problem [4.4.3]
4.5	Describe the difference in the name and text property	4.5.1	Name controls using the convention of beginning the object name with a prefix, such as btn for button or mnu for menu	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
		4.5.2	Use the text property to change the text on various object, both setting it at design time and in the code			
4.6	Describe some actions that can trigger event procedures	4.6.2	Write click event procedures	Foundation	Writing	Organizes information into an appropriate format [1.6.10]
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	4.7.1	Write program statements that change the values of properties at runtime, using Me, dot notation, equal sign, and IntelliSense	Foundation	Writing	Organizes information in an appropriate format [1.6.10]
				Thinking	Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do				ACADEMIC and WORKPLACE SKILLS What the Instruction Should Reinforce		
Knowledge		**Application**		**Skill Group**	**Skill**	**Description**
4.8	Explain the use and features of MainMenu	4.8.1	Write programs that contain a MainMenu	Foundation	Writing	Communicates thoughts, ideas, or facts in written form in a clear, concise matter [1.6.6]
				Thinking	Decision Making	Comprehends ideas and concepts related to the MainMenu function [4.2.2]
4.9	List several properties that can be used to improve the appearance of objects (Font and font features, BackColor, ForeColor, TextAlign)	4.9.1	Use these features in programs to improve the appearance of the forms and the objects on the form	Foundation	Arithmetic/ Mathematics	Comprehends mathematical ideas and concepts related to object appearance [1.1.13]
					Writing	Uses words appropriately [1.6.21]
				Thinking	Decision Making	Comprehends ideas and concepts related to object appearance [4.2.2]
4.10	Describe the purpose of a label	4.10.1	Write programs that use labels as prompts	Foundation	Speaking	Communicates thoughts, ideas, or facts in spoken form [1.5.5]
		4.10.2	Write programs that use labels to display answers	Thinking	Reasoning	Comprehends ideas and concepts related to the uses of a button [4.5.2]
4.11	Explain the use of a button	4.11.1	Write programs that use buttons and code the click event procedure	Foundation	Writing	Communicates thoughts, ideas, or facts in written form in a clear, concise matter [1.6.6]
				Thinking	Decision Making	Comprehends ideas and concepts related to the purpose of a label [4.2.2]
4.12	Explain the uses for a TextBox	4.12.1	Write programs that use text boxes for user input	Foundation	Speaking	Communicates thoughts, ideas, or facts in spoken form [1.5.5]
				Thinking	Reasoning	Comprehends ideas and concepts related to uses for a textbox [4.5.2]

Unit 5: More Visual Basic Features
Hours: 8

Terminology: AutoSize, CenterImage, CheckBox, GroupBox, Image, InputBox, MessageBox, Normal, Parameters, PictureBox, RadioButton, SizeMode, StretchImage, Visible

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do		**ACADEMIC and WORKPLACE SKILLS** What the Instruction Should Reinforce		
Knowledge	**Application**	**Skill Group**	**Skill**	**Description**
5.1 Define terminology	5.1.1 Prepare a list of terms with definitions	Foundation	Reading	Applies/Understands technical words that pertain to visual basic features [1.3.6]
			Writing	Uses words appropriately [1.6.21]
5.2 Explain the use of the checked property--for example with a RadioButton or CheckBox	5.2.1 Use the checked property in code to determine which object is selected	Foundation	Writing	Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
		Thinking	Knowing how to Learn	Applies new knowledge and skills properly [4.3.1]
5.3 Describe the features of a RadioButtons and a GroupBox	5.3.1 Write programs that use RadioButtons--code both the click event and use the checked property to determine the item selected	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
		Thinking	Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]
	5.3.2 Use a GroupBox for a set of RadioButtons			
5.4 Describe a MessageBox	5.4.1 Write code that shows a message box to display information or a warning, setting the text, caption, and icon	Foundation	Writing	Organizes information in an appropriate format [1.6.10]
		Thinking	Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]
5.5 Describe the use for a PictureBox	5.5.1 Design a form with a PictureBox with a graphic image	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
	5.5.2 Place the image in the bin folder			Organizes information in an appropriate format [1.6.10]
	5.5.3 Set the SizeMode to Normal, StretchImage, AutoSize, or CenterImage	Thinking	Reasoning	Comprehends ideas and concepts related to formatting [4.5.2]

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do				ACADEMIC and WORKPLACE SKILLS What the Instruction Should Reinforce		
Knowledge		**Application**		**Skill Group**	**Skill**	**Description**
5.6	Describe the purpose for an InputBox	5.6.1	Write programs that use an InputBox	Foundation	Speaking	Communicates thoughts, ideas, or facts in spoken form [1.5.5]
				Thinking	Reasoning	Comprehends ideas and concepts related to InputBox [4.5.2]
5.7	Describe the type of data returned by an InputBox and the needed steps to convert to the String data to numeric data	5.7.1	Write programs where the data being returned is being used as a String	Foundation	Writing	Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
		5.7.2	Write programs where the expected data is numeric, using *IsNumeric()* to check validity, and *val()* to convert to a number	Thinking	Reasoning	Comprehends ideas and concepts related to types of data [4.5.2]

Unit 6: Decision Structure
Hours: 9

Terminology: Boolean expression, Logical operators, Nested statements, Relational operator, Select case, Truth tables

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do		ACADEMIC and WORKPLACE SKILLS What the Instruction Should Reinforce		
Knowledge	**Application**	**Skill Group**	**Skill**	**Description**
6.1 Define terminology	6.1.1 Prepare a list of terms with definitions	Foundation	Reading	Applies/Understands technical words that pertain to decision structure [1.3.6]
			Writing	Uses words appropriately [1.6.21]
6.2 List relational operators	6.2.1 Write Boolean expressions that use the appropriate relational operator	Foundation	Arithmetic/ Mathematics	Interprets mathematical symbols [1.1.26]
			Writing	Applies/Uses technical words and concepts [1.6.4]
		Thinking	Reasoning	Comprehends ideas and concepts related to relational operators [4.5.2]
6.3 Describe the process of comparing two strings	6.3.1 When given two strings, determine if they are equal, the first is smaller, or the first is larger	Foundation	Writing	Communicates thoughts, ideas, or facts in written form in a clear, concise matter [1.6.6]
		Thinking	Decision Making	Comprehends ideas and concepts related to the characteristics of string data [4.2.2]
6.4 Describe a roundoff error	6.4.1 Write *if* statements that avoid the problems created by roundoff errors	Foundation	Writing	Organizes information into an appropriate format [1.6.10]
		Thinking	Problem Solving	Devises and implements a plan of action to resolve problem [4.4.3]
6.5 Explain the syntax and logic of *if* statements	6.5.1 Write programs that use *if* statements	Foundation	Writing	Organizes information into an appropriate format [1.6.10]
		Thinking	Problem Solving	Devises and implements a plan of action to resolve problem [4.4.3]
6.6 Explain the syntax and logic of *if-else* statements	6.6.1 Write statements that use if-else to make the correct decision based on the data	Foundation	Writing	Organizes information into an appropriate format [1.6.10]
		Thinking	Problem Solving	Devises and implements a plan of action to resolve problem [4.4.3]

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do				ACADEMIC and WORKPLACE SKILLS What the Instruction Should Reinforce		
Knowledge		**Application**		**Skill Group**	**Skill**	**Description**
6.7	Explain the use of logical operators *and* , *or* , and *not*	6.7.1	Write programs which require the use *and, not,* and *or*	Thinking	Reasoning	Sees relationship between two or more ideas, objects, or situations [4.5.5]
6.8	Explain the use of the *select case* statement	6.8.1	Write programs that use the *select case* statement, including cases that use a range of values, a list of possible values, and case else	Foundation	Writing	Organizes information into an appropriate format [1.6.10]
				Thinking	Problem Solving	Devises and implements a plan of action to resolve problem [4.4.3]

Unit 7: Loops
Hours: 15

Terminology: Do loop, Entrance condition loop, Exit condition loop, Flag, For-next loop, Infinite loop, Iteration, Loop, Nested loop, Sentinel

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do		**ACADEMIC and WORKPLACE SKILLS** What the Instruction Should Reinforce		
Knowledge	**Application**	**Skill Group**	**Skill**	**Description**
7.1 Define terminology	7.1.1 Prepare a list of terms with definitions	Foundation	Reading	Applies/Understands technical words that pertain to loops [1.3.6]
			Writing	Uses words appropriately [1.6.21]
7.2 Describe the purpose and syntax of for-next loops	7.2.1 Write programs that use for-next loops	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
				Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
		Thinking	Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]
7.3 Explain the procedure to use for loops to count in increments/decrements other than one	7.3.1 Write counting for loops with increments other than 1	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
				Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
		Thinking	Problem Solving	Devises and implements a plan of action to resolve problem [4.4.3]
7.4 List the types of do loops	7.4.1 Write programs that use do loops	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
				Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
		Thinking	Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do				ACADEMIC and WORKPLACE SKILLS What the Instruction Should Reinforce		
Knowledge		**Application**		**Skill Group**	**Skill**	**Description**
7.5	Describe the use of a do until loop	7.5.1	Write programs that use do…loop until	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
		7.5.2	Write programs that use do until…loop			Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
				Thinking	Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]
7.6	Describe the difference in a do unitl and a do while loop	7.6.1	Write programs that use do…loop while	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
		7.6.2	Write programs that use do while…loop			Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
				Thinking	Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]
7.7	Describe the difference in a entrance condition loop and an exit condition loop	7.7.1	Explain the potential effects of placing the condition on the do line and the loop line and conditions where each is appropriate	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
						Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
7.8	Describe a loop which ends when a sentinel or flag is entered	7.8.1	Write loops that end when a sentinel or flag is entered	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
		7.8.2	Write loops that use an InputBox and end when the Cancel button is selected			Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
				Thinking	Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]
7.9	Explain the process of using counters with loops	7.9.1	Write programs that use counters with loops	Foundation	Writing	Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
				Thinking	Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]

CAREER and TECHNICAL SKILLS What the Student Should be Able to Do		ACADEMIC and WORKPLACE SKILLS What the Instruction Should Reinforce		
Knowledge	**Application**	**Skill Group**	**Skill**	**Description**
7.10 Explain the logic of using accumulators with loops	7.10.1 Write programs that use accumulators with loops	Foundation	Writing	Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
		Thinking	Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]
7.11 Explain the difference in the effect of a while loop (entrance condition loop) and a do while (exit condition loop) loop	7.11.1 Write programs that use do while loops	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4] Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
		Thinking	Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]
7.12 Explain the syntax of nested loops	7.12.1 Determine the output of a nested loop 7.12.2 Write programs that use nested for loops	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4] Organizes information in an appropriate format [1.6.10] Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
		Thinking	Decision Making	Demonstrates decision-making skills [4.2.4]
			Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]

Glossary
Unit 1: Introduction to Programming

1. Application software – programs that perform a specific task; such as games, word processing, and spreadsheets
2. Compiler – a program that converts the entire program into machine code
3. Event-driven – a program that responds based on the user's action, such as moving or clicking the mouse or pressing a key
4. Events – an action that a user takes for which code has been written, such as a mouse click or keypress
5. Executable code – the program that has been translated into machine code
6. Hardware – the physical components of the computer
7. High-level language – programming language that is made up of English-like instructions
8. Low-level language – machine language (first-generation) and assembly language (second generation)
9. Object-Oriented Programming (OOP) – a method of programming that uses classes to create objects with associated actions and can be used repeatedly to create additional objects
10. Operating system – computer software that runs the computer; allows the user to communicate, save, print, load programs, etc.
11. Software – computer instructions, also called programs or applications
12. Source code – the program written in its original language, such as Visual Basic
13. System software – programs that run the computer and its components
14. Unicode – a digital code that assigns code numbers to characters; includes Asian and other languages with different alphabets
15. Windows application – a program that uses the Windows GUI interface

Unit 2: Programming Techniques and Characteristics of Good Programs

1. Algorithm – step-by-step process for solving a problem
2. Code – (noun) another term for program statements; (verb) to program
3. Documentation – information about a program, which includes comments inside the program as well as user's guides
4. Line continuation character – the underscore character which is used to break a program statement to another line
5. Logic errors – a problem caused by incorrect coding which produces incorrect results, rather than causing the computer to crash
6. Naming conventions – a standard for choosing identifier, such as beginning the name of a label with lbl or a button with btn
7. Program maintenance – the process of fixing errors in and updating software
8. Pseudocode – algorithm written in a combination of English and programming code; a fake or pretend program
9. Run-time error – a problem that occurs as the program is executing, which causes it to crash; such as a division by zero error
10. Statement – one instruction of a program
11. Syntax – the grammar rules of a programming language
12. Syntax errors – an error caused by breaking a language's grammar rules
13. User-friendly – a program that is easy for the user to understand and use

Unit 3: Data Types, Variables, Constants, and Mathematical Operations

1. Accumulator – a variable that is used to determine the total or sum; i.e., total = total + price
2. Boolean – a data type that can only take the values of true or false
3. Character – a single letter, symbol, digit, or punctuation mark represented in Unicode
4. Concatenation – the process of attaching the end of one string to the beginning of another string, producing a longer string
5. Constants – a named memory cell that contains a value that cannot be changed from its initial value
6. Counter – a variable that is used to keep track of the number of occurrences, such as count = count + 1
7. Data type – a description of a set of values that a variable can have; i.e., are Integer, Double, and Char
8. Dim – the reserved word that is used to declare a variable, such as Dim length as Integer
9. Fix() – a function that truncates a floating point number and works the same as Int() for whole number; for negative numbers, it returns the first negative integer that is equal to or greater; i.e., Fix(-5.4) returns -5
10. Floating point (real) – a number that has a fractional component; numbers that contain decimal portions
11. Global declaration – a variable that is declared at the beginning of the code, outside of any procedure, and which can be used in all the modules of the form; also called a module-level declaration
12. Int() – a function that returns the whole number that is equal to or less than the number; i.e., Int(5.4) returns 5 while Int(-5.4) returns -6
13. Integer – a positive or negative whole number or zero
14. Integer division – a type of division performed by the \ operator which returns the integer portion of the quotient
15. IsNumeric() function – a function that returns true if its string argument can be converted to a numeric value and false if it cannot
16. Local declaration – a variable that is declared within a procedure whose scope is limited to that procedure
17. Mathematical operators – Symbols that represent mathematical operations: +, -, *, /, \, mod, ^
18. Modulus – a type of division performed by the Mod operator which returns the remainder portion
19. Order of operations – the order that is used to perform mathematical calculations; i.e., parentheses, exponents, multiplication/division, addition/subtraction

20. Random numbers – a non-specified number that falls within a given range

21. Randomize() – a statement that is used to initialize the Rnd() function so that different random numbers are generated from run to run

22. Round-off errors – a round-off error occurs when a floating-point value cannot be stored in the alloted space and is rounded off; can produce an error when an equality comparison is made between two floating-point values

23. Static variables – a variable whose scope is local to the procedure and whose lifetime is the duration of the program

24. String – a series of characters; in Visual Basic, they are represented by string variables and string literals such as "hello"

25. Val() function – a function that is used to convert a string to a numeric value

26. Variable – a named memory location that stores a value

Unit 4: Simple Programs and Visual Basic Features

1. BackColor – the property that is used to set the background color
2. Button – an object that is commonly used for the user to click when a specific action is desired
3. Checked – the property that is true when a checkbox or radiobutton is selected
4. Control – used to create a control object that the user can interact with
5. Design view (design window) – the window where control objects are added to the form
6. Event procedure – the procedure that executes in response to an event
7. Font – the property that contains the font name, style, and size
8. ForeColor – the property that is used to set the font color
9. Form – an application interface that contains a title bar, system menu, Minimize, Maximize, and Close button as well as the objects added to the design by the programmer
10. IntelliSense – the list which is typed after the dot following Me or object name
11. Label – an object that is used to display information for the user (read-only)
12. MainMenu – the control object that places the menu on the form
13. Me – a pronoun that is used for the current form and which when followed with a dot (.) produces and intellisense list for that form
14. Name – the property that is the identifier that will be used within the program code to reference the object
15. Project – the related files in an application
16. Property – the part of a control object that defines its appearance, behavior, position, and other attributes
17. Text – the property that is the caption that will be displayed on an object
18. TextAlign – the property that is used to align the text in a label
19. TextBox – a control object that allows the user to enter a value
20. Toolbar – the row of icons below the menu bar in the IDE
21. Toolbox – the part of IDE that contains the control objects that are used to add objects to a form

Unit 5: More Visual Basic Features

1. AutoSize – sizes the picture box to fit the image
2. CenterImage – centers any image both vertically and horizontally
3. CheckBox – a type of form field that allows an Internet user to select an item by simply clicking inside a box
4. Groupbox – a control object that is used to hold a set of objects, such as a set of radio buttons
5. Image – the property that is used to add one of the images from the Resources folder to the picture box
6. InputBox – a predefined dialog box that allows the user to enter values
7. MessageBox – a predefined dialog box that displays a message for a user
8. Normal – places the image without changing its size in the upper-left corner, clipping it if necessary
9. Parameters – a value passed to a sub or function when it is invoked
10. PictureBox – a control object that is used to display an image
11. RadioButton – a set of controls where only one object can be selected from the group
12. SizeMode – the property that is used to fit the image to the picture box
13. StretchImage – sizes the image to fit the picture box, possibly distorting the image
14. Visible – the property that is set to false to hide an object

Unit 6: Decision Structure

1. Boolean expression – an expression that gives a true or false result, mainly used in repetition and selection structures

2. Logical operators – NOT, AND, OR used in boolean expressions

3. Nested statements – a statement inside another statement of the same time; i.e., an if statement inside an if statement)

4. Relational operator – several operators that are used to compare two objects, = (equal), < (less than), <= (less than or equal to), > (greater than), >= (greater than or equal to), <> (not equal)

5. Select case – selection structure whose code executes depending on the result of an expression

6. Truth tables – a complete list of all the values in a boolean expression

Unit 7: Loops

1. Do loop – a loop that continues while or until a condition becomes true

2. Entrance condition loop – a loop that tests the condition before entering the loop; therefore, may be skipped entirely (do while … loop, do until … loop, for…next)

3. Exit condition loop – a loop that tests the condition at the bottom of the loop; therefore, will always execute at least one time (do … loop until and do … loop while)

4. Flag – a condition used to signify that the loop should stop executing

5. For-next loop – a loop that counts the iteration automatically from one number to another by a specified increment

6. Infinite loop – a loop that does not end because the condition controlling it never becomes false

7. Iteration – another term for a loop; also used to refer to one pass of a loop

8. Loop – a set of statements that repeatedly perform a task

9. Nested loop – a loop inside another loop

10. Sentinel – a special value that is input to indicate the end of input