

PROGRAMMING II

Curriculum Content Frameworks

Prepared by

Xerlotta Sanders, Central High School, Little Rock
Wayne Martin, Shirley High School
Marilyn Carrell, Springdale High School

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
Ted Dean, Program Advisor
Ginger Fisher, Program Advisor
LaTrenda Jackson, Program Advisor
Office of Business/Marketing Technology
Arkansas Department of Workforce Education

Disseminated by

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

PROGRAMMING II

Suggestion: Only students with C or above in Programming I are likely to be successful. Particularly for C and lower B students, it is strongly suggested that geometry also should be required.

Depending on the language used, the terms *function*, *subprogram*, *method*, and *procedure* are similar. However, they are used somewhat differently. In some languages, the terms *subprogram* and *procedure* have the same effect as a void function in C++ and Java. In those languages, the term *function* is used only to apply to functions that return a value. (In C++ and Java, all of these are called *functions*.)

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.

The contents of these frameworks are designed to be taught in one language. The first semester of any language should be Programming I. Currently, probably the three best language choices are Visual Basic, Java, and C++. Java has an advantage over C++ since the College Board has selected it for the language of the Advanced Placement Exam, and many universities are using Java as their first programming language. Visual Basic is widely used in business programming.

The contents of these frameworks are kept to the essentials. This was done to allow the teacher time to address the specific features of the language chosen. The framework team recognizes there are vastly different additional items that need to be addressed in a visual Windows application (such as Visual Basic) rather than in a console application (used by the College Board). We expect the teacher to use the remaining time in the semester to cover those topics not listed in these frameworks.

Curriculum Content Frameworks

PROGRAMMING II

Grade Levels: 9, 10, 11, 12
Course Code: 492400

Prerequisite: Programming I
Geometry (strongly recommended)

Course Description: Programming II is a one-semester course that is a continuation of the study of the language taught in Programming I.

Table of Contents

	Page
Unit 1: Review Programming Techniques, Ethics, and Privacy	1
Unit 2: Data Validation	2
Unit 3: String Manipulation	3
Unit 4: Procedures/Subprograms/Functions with Parameters	4
Unit 5: Data Types – Boolean and Enumerated Types	5
Unit 6: One-dimensional Arrays or Vectors	6
Unit 7: Structures	8
Unit 8: Classes	9
Unit 9: Sequential Text Files	10
Glossary	11

Unit 1: Review Programming Techniques, Ethics, and Privacy

Hours: 2

Terminology: None

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 Discuss the ethical and privacy issues of programming	1.1.1 Identify ethical and privacy practices in computer programming	Personal Management	Integrity/Honesty/Work Ethic	Describes/Explains significance of integrity, honesty, and work ethics [3.2.4]	
1.2 List the steps of the programming process	1.2.1 When given an example, be able to identify the correct step	Foundation	Science	Uses equipment and techniques to solve practical problems in programming [1.4.23]	
			Writing	Organizes information in an appropriate format [1.6.10]	
		Thinking	Reasoning	Sees relationship between steps in the programming process [4.5.5]	

Unit 2: Data Validation

Hours: 5

Terminology: Data validation, Range checks

CAREER and TECHNICAL SKILLS			ACADEMIC and WORKPLACE SKILLS		
What the Student Should be Able to Do			What the Instruction Should Reinforce		
Knowledge	Application		Skill Group	Skill	Description
2.1 Explain the importance of data validation	2.1.1	Give examples of good data validation rules for a variety of situations	Foundation	Speaking	Asks questions to clarify information [1.5.3]
			Thinking	Reasoning	Comprehends ideas and concepts related to data validation [4.5.2]
2.2 Explain the logic of numeric range checks	2.2.1	Write programs that use range checks	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
			Thinking	Decision Making	Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19] Demonstrates decision-making skills [4.2.4]
2.3 Explain the logic of data validation to match a particular pattern	2.3.1	Write programs that require data to fit a specified pattern (i.e., Social Security Number 123-45-6789, phone number 123-456-7890, etc.)	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
			Thinking	Reasoning	Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19] Applies rules and principles to a new situation [4.5.1]

Unit 3: String Manipulation

Hours: 10

Terminology: American Standard Code for Information Interchange (ASCII), Concatenation, Unicode

CAREER and TECHNICAL SKILLS		ACADEMIC and WORKPLACE SKILLS		
What the Student Should be Able to Do		What the Instruction Should Reinforce		
Knowledge	Application	Skill Group	Skill	Description
3.1 Explain the syntax and features of various commands dealing with ASCII or Unicode numbers and their corresponding characters	3.1.1 Write program lines to determine the ASCII number of a character	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
	3.1.2 Write programs to use the ASCII number to print the corresponding character	Thinking	Reasoning	Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19] Sees relationship between two or more ideas, objects, or situations [4.5.5]
3.2 Explain the syntax and purpose of commands that handle all or part of a string and that concatenate strings	3.2.1 Write programs to determine the number of characters in a string	Foundation	Writing	Applies/Uses technical words and concepts [1.6.4]
	3.2.2 Write programs to print a particular group of characters that are contained in a string	Thinking	Creative Thinking	Uses language, style, organization, and format appropriate to subject matter, purpose, and audience [1.6.19]
	3.2.3 Write programs that concatenate multiple strings into one			Uses imagination to create something new [4.1.1] Combines ideas or information in a new way [4.1.2]
3.3 Explain the reasons why breaking a string into its component parts is important	3.3.1 Write code that will take the first part of a string from a longer string (i.e., taking the area code from a telephone number)	Foundation	Writing	Sees relationship between two or more ideas, objects, or situations [4.5.5]
	3.3.2 Write code that will take characters from the middle of a string (i.e., removing the middle name from the full name)	Thinking	Problem Solving	Organizes information in an appropriate format [1.6.10]
	3.3.3 Write code that will take characters from the right side of a string (i.e., ZIP code from the address)			Devises and implements a plan of action to resolve problems [4.4.3]

Unit 4: Procedures/Subprograms/Functions with Parameters

Hours: 15-10

Terminology: Argument, Parameter, Reference parameter, Value parameter

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 Explain the difference between <i>argument</i> and <i>parameter</i>	4.1.1 Give examples of arguments and parameters	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to arguments and parameters [4.5.2]	
4.2 Explain the matching of arguments in the function call to the function parameters	4.2.1 Write programs that use arguments in function/procedure calls	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to arguments and parameters [4.5.2]	
4.3 Explain when to use value parameters	4.3.1 Write functions/procedures with value parameters	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to value parameters [4.5.2]	
4.4 Explain when to use reference parameters	4.4.1 Write functions/procedures with reference parameters	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to reference parameters [4.5.2]	
4.5 Explain when to use functions that return a value	4.5.1 Write functions that return values	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to returning a value [4.5.2]	
4.6 Explain why array parameters and other data structure parameters should be passed by reference	4.6.1 Write functions that have array or other data structure parameters	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to value parameters [4.5.2]	
4.7 Explain when to use a constant reference parameter (in languages where available)	4.7.1 Write functions that have constant reference parameters (in languages where available)	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to reference parameters [4.5.2]	

Unit 5: Data Types – Boolean and Enumerated Types

Hours: 5

Terminology: Boolean, Boolean expressions, Enumerated type

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 Explain when and where to use Boolean expressions and variables	5.1.1 Write programs that use Boolean expressions and variables	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to Boolean expressions and values [4.5.2]	
5.2 Discuss enumerated type (i.e., color, cards, etc.)	5.2.1 Write programs that declare and use enumerated types	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]	
5.3 Explain the purpose of using enumerated types	5.3.1 Give examples of when enumerated types are needed	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to enumerated types [4.5.2]	

Unit 6: One-dimensional Arrays or Vectors

Hours: 20

Terminology: Array, Array elements, Array initialization, Dimensions, Index, Parallel arrays, Subscript, Vector

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 Explain arrays and vectors	6.1.1 Give examples of when the use of arrays or vectors is appropriate	Foundation	Writing	Presents answers/conclusions in a clear and understandable format [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to arrays or vectors [4.5.2]	
6.2 Explain the use of dimensions and subscripts and the syntax of commands to use them	6.2.1 Write an appropriate program to dimension one-dimensional arrays or vectors	Foundation	Writing	Organizes information in an appropriate format [1.6.10]	
	6.2.2 Use subscript to access particular elements in a one-dimensional array or vector	Thinking	Problem Solving	Presents answers/conclusions in a clear and understandable format [1.6.13] Devises and implements a plan of action to resolve problems [4.4.3]	
6.3 Explain the logical steps in initializing and loading a one-dimensional array	6.3.1 Write program lines that initialize a one-dimensional array or vector	Foundation	Writing	Comprehends ideas and concepts related to subscripts [4.5.2]	
	6.3.2 Write program lines that read data from a file into a one-dimensional array or vector	Thinking	Problem Solving	Organizes information in an appropriate format [1.6.10] Devises and implements a plan of action to resolve problems [4.4.3]	
6.4 Explain the logical steps in traversing a one-dimensional array to perform calculations and comparisons	6.4.1 Write loops that traverse a one-dimensional array or vector, performing calculations and comparisons	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]	
6.5 Explain the logical steps in printing an entire array of data	6.5.1 Write loops that print the contents of a one-dimensional array or vector	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	
6.6 Explain the logical steps to insert a value in a one-dimensional array or vector	6.6.1 Write code that inserts values into an existing array or vector	Foundation Thinking	Writing Problem Solving	Organizes information in an appropriate format [1.6.10] Devises and implements a plan of action to resolve problems [4.4.3]	
6.7 Explain the logical steps in deleting elements in a one-dimensional array or vector	6.7.1 Write code that deletes elements from an existing array or vector	Foundation Thinking	Writing Problem Solving	Organizes information in an appropriate format [1.6.10] Devises and implements a plan of action to resolve problems [4.4.3]	
6.8 Explain the use of parallel one-dimensional arrays or vectors	6.8.1 Write programs that contain parallel one-dimensional arrays or vectors	Foundation Thinking	Writing Problem Solving	Organizes information in an appropriate format [1.6.10] Devises and implements a plan of action to resolve problems [4.4.3]	

Unit 7: Structures

Hours: 5

Terminology: Structure

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 Discuss structure	7.1.1 Give an example of a useful structure	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to structures [4.5.2]	
7.2 Explain the advantages of structure in handling large amounts of related data	7.2.1 Write a program that uses a structure	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to structures [4.5.2]	
7.3 Explain the process of traversing an array/vector of structures to process data	7.3.1 Write functions/procedures that traverse an array of structures to process data	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]	
7.4 Explain the logical steps to insert a value in a one-dimensional array or vector of structures	7.4.1 Write code that inserts values in an existing array or vector of structures	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]	
7.5 Explain the logical steps in deleting elements in a one-dimensional array or vector of structures	7.5.1 Write code that deletes elements from an existing array or vector of structures	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]	

Unit 8: Classes

Hours: 5

Terminology: Class, Instantiation, Member method (member function), Object

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	
8.1 Discuss class	8.1.1 Give an example of a class used in this unit	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to classes [4.5.2]	
8.2 Explain the advantages of using classes	8.2.1 Give an example of a useful class	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to classes [4.5.2]	
8.3 Discuss object	8.3.1 Give examples of data that would be contained in example classes and the methods that would be needed to manipulate that object	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to objects [4.5.2]	
8.4 Discuss instantiation	8.4.1 Write a program that instantiates an object of the class	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]	
8.5 Explain member methods	8.5.1 Write a program that uses a member of the class	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]	

Unit 9: Sequential Text Files

Hours: 15

Terminology: Append, End of file (EOF), File, Random access, Sequential access, Text file

CAREER and TECHNICAL SKILLS		ACADEMIC and WORKPLACE SKILLS			
What the Student Should be Able to Do		What the Instruction Should Reinforce			
Knowledge	Application	Skill Group	Skill	Description	
9.1 Discuss files	9.1.1 Give examples of when using files would be useful	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to files [4.5.2]	
9.2 Explain the difference between sequential and random access of files	9.2.1 List the advantages of using sequential files vs. random-access files	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
		Thinking	Reasoning	Comprehends ideas and concepts related to files [4.5.2]	
9.3 Explain the difference between opening text files for output, append, and input	9.3.1 Write programs that open files appropriately for output, append, and input	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]	
9.4 Explain the logic of reading data sequentially from a text file	9.4.1 Write programs that read files	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]	
9.5 Explain the terminator characters at the end of lines and end of the file	9.5.1 Write programs that read the data until the end of file (EOF)	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]	
9.6 Explain the logic of writing data sequentially to a text file	9.6.1 Write programs that write data sequentially to a text file	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]	
9.7 Explain the function of closing the file	9.7.1 Close files in all programs using files	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]	

Glossary

Unit 1: Review Programming Techniques, Ethics, and Privacy

No terminology for this unit

Unit 2: Data Validation

1. Data validation – the examination of data within a program prior to its use to determine whether it fits the criteria of valid data for that situation
2. Range checks – used to check that numbers are within a prescribed range

Unit 3: String Manipulation

1. American Standard Code for Information Interchange (ASCII) – a numerical code that represents a character as a value; there are 128 common codes
2. Concatenation – the process of appending a string with parts of another string
3. Unicode – a 16-bit digital code that represents every letter of an alphabet and symbols of every culture

Unit 4: Procedures/Subprograms/Functions with Parameters

1. Argument – the entries within parentheses that are supplied to a function
2. Parameter – a variable listed in a function or procedure heading that receives data from the calling statement or that receives and returns values to the calling statement
3. Reference parameter – a parameter that can both receive values from the calling statement and return values to the calling statement
4. Value parameter – a parameter that can only receive values from the calling statement

Unit 5: Data Types – Boolean and Enumerated Types

1. Boolean – a data type that stores logical values of true or false
2. Boolean expression – an expression that returns the logical value of true or false; i.e., if (age>=18) or while (counter <25)
3. Enumerated type – a data type that defines a related set of named constants

Unit 6: One-dimensional Arrays or Vectors

1. Array – a data type that can store more than a set of values (list of values) of the same type, which can be accessed by an index
2. Array elements – pieces of data (values) or members that make up the array
3. Array initialization – the setting of all the values of an array to a predetermined value, such as zero for numeric arrays or the null (empty) string for strings
4. Dimensions – a table with several rows and one column is one-dimensional; tables that have more than one row and more than one column are two-dimensional
5. Index – a number used to access the elements of an array, vector, matrix, etc.; also called a subscript
6. Parallel arrays – arrays that use the same index value to indicate related values stored in separate arrays
7. Subscript – a number used to access the elements of an array, vector, matrix, etc.; also called an index
8. Vector – a class, or structure, that can dynamically expand to hold a list of data or objects that are accessed by a single subscript

Unit 7: Structures

1. Structure – a data type that is made up of a collection of elements

Unit 8: Classes

1. Class – the definition (description of attributes and behavior) of the variables within a specific object
2. Instantiation – to declare an object of a class
3. Member method (member function) – a function implemented within a class definition that accesses or manipulates the data members of that class
4. Object – a unique instance of a class containing the data and functions that manipulate the data

Unit 9: Sequential Text Files

1. Append – to attach or add to as a supplement at the end
2. End of file (EOF) – a special marker used to indicate that the end of a file has been reached
3. File – a collection of text, graphics, or any combination thereof stored in secondary storage and assigned a file name
4. Random access – a file in which each record is the same length and can be accessed in any order
5. Sequential access – a type of file that must be accessed from beginning to end
6. Text file – a file that contains only American Standard Code for Information Interchange (ASCII) characters