

PROGRAMMING II - JAVA

Curriculum Content Frameworks

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

Prepared by

Carl Frank, Arkansas School for math and Science
Linda Joplin, Southside High School, Ft. Smith
Renee Nichols, Rogers High School
Gini Cocanower, Bentonville 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
Ginger Fisher, Program Advisor
LaTrenda Jackson, Program Advisor
Tim Johnston, 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 - JAVA

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 be a prerequisite for this course.

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 designed to be taught in one language. The first semester of any language should be Programming I. As of the writing of this framework 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.

As with all frameworks, only the essentials are covered. Teachers will, therefore, have 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 - JAVA

Grade Levels: 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: Classes and Objects	5
Unit 5: Methods	6
Unit 6: One-dimensional Arrays or Vectors	7
Unit 7: ArrayLists, Searching and Sorting	9
Glossary	10

Unit 1: Review Programming Techniques, Ethics, and Privacy

Hours: 2

Terminology: None

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		
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	Identify correct step when given an example	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 Define terminology	2.1.1 Manipulate vocabulary and concepts individually and in groups	Foundation	Reading	Applies information and concepts derived from printed materials [1.3.3] Applies/Understands technical words that pertain to data validation [1.3.6]	
2.2 Explain the importance of data validation	2.2.1 Give examples of good data validation rules for a variety of situations	Foundation Thinking	Speaking Reasoning	Asks questions to clarify information [1.5.3] Comprehends ideas and concepts related to data validation [4.5.2]	
2.3 Explain the logic of numeric range checks	2.3.1 Write programs that use range checks	Foundation Thinking	Writing Decision Making	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] Demonstrates decision-making skills [4.2.4]	
2.4 Explain the logic of data validation to match a particular pattern	2.4.1 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 Thinking	Writing Reasoning	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] 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 Define terminology	3.1.1 Manipulate vocabulary and concepts individually and in groups	Foundation	Reading	Applies information and concepts derived from printed materials [1.3.3] Applies/Understands technical words that pertain to string manipulation [1.3.6]
3.2 Explain the syntax and features of various commands dealing with ASCII or Unicode numbers and their corresponding characters	3.2.1 Write program lines to determine the ASCII number of a character	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]
	3.2.2 Write programs to use the ASCII number to print the corresponding character	Thinking	Reasoning	Sees relationship between two or more ideas, objects, or situations [4.5.5]
3.3 Explain the syntax and purpose of commands that handle all or part of a string and that concatenate strings	3.3.1 Write programs to determine the number of characters in a string	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]
	3.3.2 Write programs to print a particular group of characters that are contained in a string	Thinking	Creative Thinking	Uses imagination to create something new [4.1.1] Combines ideas or information in a new way [4.1.2]
	3.3.3 Write programs that concatenate multiple strings into one			Reasoning

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
3.4 Explain the reasons why breaking a string into its component parts is important	3.4.1 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) 3.4.2 Write code that will take characters from the middle of a string (i.e., removing the middle name from the full name) 3.4.3 Write code that will take characters from the right side of a string (i.e., ZIP code from the address)	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 4: Classes and Objects

Hours: 10-15

Terminology: Attribute, Behavior, Class, Constructor, Default constructor, Instance variables, Instantiation, Method, New operator, Object, Overloaded constructor, Return type, State, Visibility modifier

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	
4.1 Define terminology	4.1.1 Manipulate vocabulary and concepts individually and in groups	Foundation	Reading	Applies information and concepts derived from printed materials [1.3.3] Applies/Understands technical words that pertain to classes and objects [1.3.6]	
4.2 Explain the advantages of using classes	4.2.1 Define real-world entities in terms of state and behavior. Examples: bank accounts, employee records, or graphical shapes	Thinking	Decision Making	Comprehends ideas and concepts related to designing a class [4.2.2]	
			Problem Solving	Devises and implements a plan of action to solve a problem [4.4.3]	
4.3 Explain the format of simple classes	4.3.1 Write a class that contains data members and methods	Thinking	Decision Making	Comprehends ideas and concepts related to designing a class [4.2.2]	
	4.3.2 Write a driver class that instantiates an object		Problem Solving	Devises and implements a plan of action to solve a problem [4.4.3]	
4.4 Explain the visibility modifiers <i>public</i> and <i>private</i>	4.4.1 Write a class that has private data members and public methods	Foundation	Reading	Applies/understands technical words that pertain to visibility modifiers [1.3.6]	
		Thinking	Reasoning	Applying a rule or principle to a new situation [4.5.1]	
4.5 Explain constructor methods	4.5.1 Write a class that contains a constructor with no arguments	Foundation	Writing	Organizes information in appropriate format [1.6.10]	
	4.5.2 Write a class that contains a constructor with one or more arguments	Thinking	Knowing How to Learn	Applies new knowledge and skills properly [4.3.1]	
	4.5.3 Write a class that overloads the constructor				

Unit 5: Methods

Hours: 10-15

Terminology: Accessor method, Actual parameters, Aliasing, Argument, Formal parameter, Mutator method, Pass by reference, Pass by value

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	
5.1 Define terminology	5.1.1 Manipulate vocabulary and concepts individually and in groups	Foundation	Reading	Applies information and concepts derived from printed materials [1.3.3] Applies/Understands technical words that pertain to methods [1.3.6]	
5.2 Explain the difference between formal and actual parameters	5.2.1 Give examples of formal and actual parameters	Foundation Thinking	Writing Reasoning	Presents answers/conclusions in a clear and understandable form [1.6.13] Comprehends ideas and concepts related to arguments and parameters [4.5.2]	
5.3 Explain the matching of actual parameters to formal parameters in the method call	5.3.1 Write programs that use parameters in method calls	Foundation Thinking	Writing Reasoning	Presents answers/conclusions in a clear and understandable form [1.6.13] Comprehends ideas and concepts related to arguments and parameters [4.5.2]	
5.4 Explain the difference between passing primitive data types as parameters and passing objects as parameters	5.4.1 Write methods passing both primitive data types and String objects	Foundation Thinking	Writing Reasoning	Presents answers/conclusions in a clear and understandable form [1.6.13] Comprehends ideas and concepts related to value parameters [4.5.2]	
5.5 Differentiate between void and value returning methods	5.5.1 Write methods that return values (accessor or "getter" methods) 5.5.2 Write void methods (mutator or "setter" methods)	Foundation Thinking	Writing Reasoning	Presents answers/conclusions in a clear and understandable form [1.6.13] Comprehends ideas and concepts related to returning a value [4.5.2]	
5.6 Explain how to pass arrays, ArrayLists, and other objects as parameters	5.6.1 Write methods that have arrays, arraylists or other objects as parameters	Foundation Thinking	Writing Reasoning	Presents answers/conclusions in a clear and understandable form [1.6.13] Comprehends ideas and concepts related to value parameters [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 Define terminology	6.1.1 Manipulate vocabulary and concepts individually and in groups	Foundation	Reading	Applies information and concepts derived from printed materials [1.3.3] Applies/Understands technical words that pertain to arrays and vectors [1.3.6]
6.2 Explain the logical steps in initializing and loading a one-dimensional array	6.2.1 Write an appropriate program to declare, instantiate and initialize a one-dimensional array	Foundation	Writing	Organizes information in an appropriate format [1.6.10] Presents answers/conclusions in a clear and understandable format [1.6.13]
		Thinking	Problem Solving	Devises and implements a plan of action to resolve problems [4.4.3]
			Reasoning	Comprehends ideas and concepts related to subscripts [4.5.2]
6.3 Explain the use of subscripts and the syntax to use them	6.3.1 Use subscript to access particular elements in a one-dimensional array	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]
6.4 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, 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	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		ACADEMIC and WORKPLACE SKILLS			
What the Student Should be Able to Do		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	6.6.1 Write code that inserts values into an existing array	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	6.7.1 Write code that deletes elements from an existing array	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: ArrayLists, Searching and Sorting

Hours: 5

Terminology: ArrayList, Insertion sort, Selection sort, Sequential search, Binary search, Traverse

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 Manipulate vocabulary and concepts individually and in groups	Foundation	Reading	Applies information and concepts derived from printed materials [1.3.3] Applies/Understands technical words that pertain to ArrayLists, searching and sorting [1.3.6]	
7.2 Discuss the difference between an ArrayList and an array	7.2.1 Determine whether an array or ArrayList would be the best data structure for a given scenario	Foundation	Writing	Presents answers/conclusions in a clear and understandable form [1.6.13]	
	7.3.2 Discuss the differences between arrays and ArrayLists when adding and deleting	Thinking	Reasoning	Comprehends ideas and concepts related to structures [4.5.2]	
	7.2.3 Discuss the differences between arrays and ArrayLists when accessing elements				
7.3 Know the difference between type specific and generic ArrayLists	7.3.1 Declare and instantiate both generic and type specific ArrayLists	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]	
7.4 Know how to add, remove, and retrieve elements from an ArrayList	7.4.1 Write a program that adds, removes, and retrieves elements from an ArrayList	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 difference between a binary search and a sequential search	7.5.1 Write programs to demonstrate binary and sequential searches	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]	
7.6 Know the difference between insertion, selection, and merge sorts	7.6.1 When given code identify the type of sort used	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 validity for that situation
2. Range checks – a check to determine 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: Classes and Objects

1. Attribute – see instance variable
2. Behavior – what an object does as defined by its methods
3. Class – the blueprint of an object; the model that defines the variables and methods an object will contain when instantiated
4. Constructor – a method that initializes a newly instantiated object
5. Default constructor – a method that Java provides for creating objects of a class in the event a constructor has not been explicitly defined in the class definition
6. Instance variables – a variable defined in a class for which every object of the class has its own value
7. Instantiation – the act of creating an object from a class
8. Method – a sequence of statements that has a name, may have parameters, and may return a value
9. New operator – a Java reserved word that is also an operator, used to instantiate an object from a class
10. Object – an instance of a class
11. Overloaded constructor – a constructor method that contains a different type or number of parameters
12. Return type – the type of value returned from a method
13. State – the state of being of an object defined by the values of its data members
14. Visibility modifier – a key word such as "public" or "private" that indicates the accessibility of a feature; also known as access modifier

Unit 5: Methods

1. Accessor method – a method used to examine an attribute of an object without changing it, aka a "getter" method
2. Actual parameters – the entries within parentheses that are supplied to a method in the method statement
3. Aliasing – when more than one reference refers to the same object
4. Argument – another name for actual parameter – the entries within parentheses that are supplied to a method
5. Formal parameter – a variable listed in a method heading that receives data from the method call
6. Mutator method – a method used to change the value of an attribute of an object
7. Pass by reference – a method of passing parameters that causes any change to the formal parameter of a method to change the actual parameter, which is its alias, in the calling statement
8. Pass by value – a method of passing parameters where the formal parameter receives a copy of the actual parameter; therefore a change to the formal parameter does not affect the actual parameter

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: ArrayLists, Searching and Sorting

1. ArrayList – a Java class that implements a growable array of objects
2. Insertion sort – a sorting algorithm in which each value is inserted, one at a time, into a sorted subset of the entire list
3. Selection sort – a sorting algorithm in which each value is inserted, one at a time, is placed in its final sorted position
4. Sequential search – a search algorithm in which each item is compared to the target value until the target is found or the list is done
5. Binary search – a search algorithm that compares the middle element of a sorted list to the target value, halving the scope of the search each time
6. Traverse – visiting each element of an array or an ArrayList