Fundamentals of Programming and Software Development	Scope and Sequence
Unit Lesson	**Objectives**
INTRODUCTION TO COMPUTERS	
Computer History	
	Describe the history and evolution of the computer.
	Identify major contributors and their influence on the field of computer technology.
	Create a timeline illustrating the most significant contributions to computing technology.
	Discuss the past and present impact of computer technology on our lives.
Project: Computer Generations	
Introduction to Computer Hardware	
	Identify computer hardware components and their functions.
	Differentiate between storage and memory.
	Differentiate between input and output devices.
	Discuss the differences between random-access memory (RAM) and read-only memory (ROM).
	Discuss the function of the central processing unit.
Project: Understanding Hardware	
Introduction to Computer Software	
	Differentiate between system software and application software.
	Name and describe several programming languages.
	Explain the difference between high-level language and low-level language.
	Match popular software names with the appropriate type and intended use.
	Describe an input process output diagram and its function.
Design and Function of the Central Processing Unit	

Fundamentals of Programming and Software Development		Scope and Sequence
Unit	**Lesson**	**Objectives**
		Explain the function of the control unit (CU).
		Explain the function of the arithmetic logic unit (ALU).
		Explain how cache works in the CPU.
		Describe the importance of following Order of Operations in the ALU.
		List the correct PEMDMAS order of operations.
		Describe the role of logic gates.
	Introduction to Java Programming	
		Discuss how an IPO chart and pseudocode improve program logic.
		Explain how to write pseudocode for a simple, everyday task.
		Explain basic Java syntax rules.
		Define Java objects, attributes, methods, and values.
		Explain how to write and compile a simple Java program.
	Project: Writing Your First Java Program	
	Java Syntax Overview	
		Explain how to construct a Java programming template.
		Identify the core components required for every Java program.
		Describe the syntax for creating comments and strings.
		Define the central keywords used by Java.
	Project: Hello World! Documentation	
	Test	
JAVA		
	Introduction to Java Variables	

Fundamentals of Programming and Software Development		Scope and Sequence
Unit	**Lesson**	**Objectives**
		Describe how to construct variable names in accordance with Java syntax and standard programming practices.
		Explain how to declare and initialize variables.
		Discuss the performance of simple math operations using variables and Java symbols.
	Project: Using Variables in Java	
	Java Math Operations	
		Understand the difference between the assignment operator and the equal-to comparison operator.
		Identify the Java comparison operators, and describe their functions.
		Describe the value limits for each Java data type.
		Perform mathematical and comparison operations using variables.
		Create, compile and run a Java program that declares, initializes, and manipulates, and that displays variable names and values.
	Project: Using Mathematical Operators in Java	
	Operators and Escape Sequences	
		Write single-line syntax for a conditional operator statement in Java.
		Explain the if-then-else decision process behind conditional operators.
		Identify the correct escape sequence to achieve desired output formatting.
		Create a Java program that utilizes conditional operators to compare data values and make decisions based on results.
	New Data Types and the If Statement	
		Demonstrate understanding of the use of char and string data types.
		Distinguish the difference between char and string data.
		Compare the logical foundations of the if statement and if-else statement.

Fundamentals of Programming and Software Development		Scope and Sequence
Unit	**Lesson**	**Objectives**
		Describe the purpose of the System.in.read(); statement.
		Explain how to write Java code using if statements, if-else statements, and System.in.read() statements.
	Project: Using If and If-Else Statements and Reading User Input	
	Switch and Case	
		List several best practices for coding programs.
		Describe the function of the switch-case statement.
		Explain the distinction between the keywords break and default.
		Identify approriate situations for using the switch-case and nested if statements.
		Write efficient code using switch-case and nested if statements.
	Project: Using Switch-Case and Nested If Statements	
	User-Defined Methods	
		Describe the value of user-defined methods in Java programming.
		Define the keywords public, private, and void, and describe their purpose.
		Write user-defined methods that accept parameters and return values..
		Discuss the importance of acting responsibly in the computer workplace environment.
	Test	
PROGRAMMING		
	Introduction to the For Loop	
		Describe the function and purpose of a for loop.
		Identify the components of a for loop structure.
		Differentiate and describe postfix and prefix operators.

Fundamentals of Programming and Software Development		Scope and Sequence
Unit	**Lesson**	**Objectives**
		Write a Java program implementing a for loop.
	Project: Grading on a Loop	
	Loops–Practice with the Do-While Loop	
		Describe the function and purpose of a do-while loop.
		Identify the components of a do-while loop structure.
		Differentiate between the appropriate use of the for loop and the do-while loop.
		Describe the cause of an infinite loop.
		Explain the concept of debugging a computer program.
	Loops–Practice with the While Loop	
		Describe the function and purpose of a while loop.
		Identify the components of a while loop structure.
		Discuss key distinctions among the for loop, do-while loop, and while loop.
		Determine the most appropriate loop structure to use for a particular situation.
		Explain how to create pseudocode and a Java program that contains an appropriate loop.
		Explain how to create pseudocode and a Java program that contains an appropriate loop.
	Project: Using Loops in a Guessing Game	
	Arrays–Syntax and Use	
		Explain the logic and rationale behind Java arrays.
		Identify the syntax components to create and populate an array.
		Declare an array and populate it with values.
		Compare the long- and short-form techniques to declare an array.
	Arrays–Passing by Reference	

Fundamentals of Programming and Software Development		Scope and Sequence
Unit	**Lesson**	**Objectives**
		Differentiate and describe the concepts of passing by reference and passing by value.
		Explain the concept of an array index.
		Describe and employ two techniques to declare and populate arrays in Java.
		Create arrays for personal data using appropriate Java syntax.
	Project: Professional Associations Research	
	Parallel and Multidimensional Arrays	
		Explain the logic and structure of two- and three-dimensional arrays.
		Describe the construction of multiple arrays using proper syntax.
		Justify the use of a multidimensional array logic model for a real-world data situation.
		Describe the concept of desk checking a program.
	Project: The Logic of Multidimensional Arrays	
	Test	
ADVANCED PROGRAMMING		
	Classes and Objects	
		Discuss the relationship of Java classes, variables, and methods.
		Explain how to construct classes and related objects using correct Java syntax.
		Explain the distinctions between public and private as well as abstract and concrete classes.
		Define the concept and role of importing packages in Java.
		Formulate clear opinions on the role of usability in the computer technology industry.
	Project: The Importance of Usability	
	Constructors and Packages	
		Discuss the concept and syntax of classes.

Fundamentals of Programming and Software Development		Scope and Sequence
Unit	**Lesson**	**Objectives**
		Explain the use of a Java-defined constructor to instantiate a class object.
		Describe a user-defined constructor.
		Compare and contrast Java-defined constructors with user-defined constructors.
		Explain the use of classes to produce packages.
	Project: Creating Packages	
	Flowcharts Mapping	
		Identify the correct flowchart symbol for specific program components and actions.
		Interpret and explain the logic and flow of control depicted in a flowchart.
		Design a flowchart from an IPO chart and pseudocode.
		Construct flowchart sequences to depict if-then and looping scenarios.
	HTML Basics	
		Describe the function of tagging in HTML.
		Identify basic HTML tags and describe their purpose.
		Create a basic HTML file that can be viewed in a web browser.
	Project: A Web Page Essay About the Web	
	HTML Images, Links, and Web Development Tools	
		Describe how to construct list formats in two formats using HTML.
		Explain how to write HTML code for a web page that includes images.
		Describe how to construct a web page with links to external sites.
		Define WYSIWYG and explain its benefits in developing web pages.
	Project: Your Favorite Recipe–On a Web Page	
	Event-Driven Programming and Visual Basic	

Fundamentals of Programming and Software Development		Scope and Sequence
Unit	**Lesson**	**Objectives**
		Define and describe a graphical user interface.
		Explain the logic of event-driven programming.
		Identify user-initiated events on a computing device interface.
		Describe the relationship between controls, events, and subroutines.
	Test	
GUI PROGRAMMING AND WEB APPLICATIONS		
	Software Development Life Cycle	
		Identify and describe the stages of the software development process.
		Explain the roles of clients and end users in the software development process.
		Compare and contrast major software development life-cycle models.
		Choose appropriate software development models based on various project characteristics.
	Project: Planning a Software Development Project	
	Programming Languages	
		Identify major categories of programming languages.
		Describe primary characteristics of popular programming languages.
		Classify popular programming languages into key categories.
	User-Centered Software Design	
		Explain the purpose and value of user-centered design.
		Distinguish between contextual design and participatory design approaches.
		List examples of “form over function” in existing products.
		Create an app protoype that demonstrates basic principles of usability.
	Project: User-Testing a Product Prototype	

Fundamentals of Programming and Software Development		Scope and Sequence
Unit	**Lesson**	**Objectives**
	Skills and Interests for Software Careers	
		Identify the technical skills necessary to be a computer programmer.
		Identify and describe transferable skills necessary to succeed as a computer programmer.
		Describe the Strong Interest Inventory themes and formulate a personal Holland code.
	Project: Taking Stock	
	Software Industry Careers	
		Distinguish the roles and responsibilities of computer programmers and software developers.
		Explain the difference between application software developers and system software developers.
		Describe education requirements for programmers, software developers, and web developers.
	Project: Planning Your Computer Science Degree Program	
	New Trends and Technologies	
		Define the concept of the Web of Things and Internet of Things.
		Explain how the spread of computing devices is changing expectations for software.
		Analyze how Big Data presents opportunities and challenges.
		Compare and contrast functional programming and probabilistic modeling.
	Test	
COURSE PROJECT, REVIEW, AND EXAM		
	Course Project Part 6: Issues and Experiences in the World of Software Development	
	Review	
	Exam	