Sun Educational Services – Instructor-Led Course Description

Fundamentals of Java

SL-110

The Fundamentals of the Java course provides students, with little or no programming experience, with the basics of programming using the Java programming language. This course teaches the significance of object-oriented programming, the keywords and constructs of the Java programming language, and the steps required to create simple Java technology programs. The course starts with teaching basic object-oriented (OO) concepts and object-oriented analysis and design practices as they relate to Java technology. It is a preparatory course so that students can learn the necessary background in preparation for taking SL-275: *Java Programming Language*.

Who can benefit

Students who can benefit from this course are individuals who have basic mathematical, logical, and analytical problem-solving skills and who want to begin learning the Java programming language. These students include technical writers, Web developers, technical managers, and individuals with a technical, non-programming background, such as system administrators. This course is also appropriate for novice programmers and those programmers who prefer to start learning the Java programming language at an introductory level. Individuals that have had some programming experience with a non-object-oriented programming background, such as traditional C developers, will benefit from this course because it helps them to begin the migration to the Java programming language.

Prerequisites

To succeed fully in this course, students should be able to:

- Create and edit text files using a text editor
- Use a World Wide Web (WWW) browser
- Solve logic problems
- Describe the concept of a variable
- Execute commands using a command-line interface

- Have at minimum of programming experience
- Have a minimum experience in designing applications

Skills Gained

Upon completion of this course, students will be able to:

- Identify objects in a problem domain
- Design a class diagram for a problem domain
- Develop basic Java classes based on a class diagrams
- Identify Java technology product groups and list the stages of the product life cycle
- Analyze a problem and design classes to solve the problem
- Develop and test a Java technology application
- Use primitive data types to store data within variables
- Demonstrate object-oriented reuse principles through the use of object references
- Use operators and decision constructs
- Use loop constructs
- Develop and use methods
- Implement encapsulation and constructors
- Create and use one-dimensional and multidimensional arrays
- Implement inheritance to extend classes
- Demonstrate understanding of key object-oriented programming concepts

Related courses

After:

- SL-275: *Java Programming Language*
- OO-226: *Object-Oriented Application Analysis and Design for Java Technology* (UML)

Course Outline

Module 01 : Objects

- Describe abstraction and how it is used in object orientation
- Identify objects and non-objects from a problem domain
- Describe object encapsulation

Module 02 : Classes

- Group objects with similar attributes and common operations in classes
- Explain how classes are used to define objects
- Define inheritance and explain how it relates to software reuse
- Define generalization and specialization and how they relate to inheritance
- Define polymorphism and explain how inheritance promotes polymorphism
- Define abstract classes

Module 03 : Object Interaction

- Explain how objects interact with each other through object messaging
- Define association and composition
- Decide whether a relationship between two objects should be association or composition
- Define the lifetime of an object with regard to association and composition
- Define the custody of an object with regard to association and composition

Module 04 : Object-Oriented Analysis and Design Using UML

- Create a set of use cases to describe a problem domain
- Create a sequence diagram for a use case
- Create a class diagram for a problem domain
- Create an activity diagram for a use case
- Code class declarations for the class diagram

Module 05 : Explaining Java Technology

- Describe key concepts of the Java programming language
- List the three Java technology product groups
- Summarize each of the seven stages of the product life cycle primitive variables

Module 06 : Analyzing a Problem and Designing a Solution

- Analyze a problem using object-oriented analysis
- Design classes from which objects will be created

Module 07 : Developing and Testing a Java Technology Program

- Identify the four components of a class in the Java programming language
- Use the main method in a test class to run a Java technology program from the command line
- Compile and execute a Java technology program

Module 08: Declaring, Initializing, and Using Variables

- Identify the uses for variables and define the syntax for a variable
- List the eight Java programming language primitive data types
- Declare, initialize, and use variables and constants according to Java programming language guidelines and coding standards
- Modify variable values using operators
- Use promotion and type casting

Module 09: Creating and Using Objects

- Declare, instantiate, and initialize object reference variables
- Compare how object reference variables are stored in relation to primitive variables
- Use a class (the String class) included in the Java Software Developer Kit (SDK)
- Use the Java 2 Platform, Standard Edition (J2SE) class library specification to learn about other classes in this application programming interface (API)

Module 10: Using Operators and Decision Constructs

- Identify relational and conditional operators
- Create *if* and *if/else* constructs
- Use the *switch* construct

Module 11: Using Loop Constructs

- Create *while* loops
- Develop *for* loops
- Create *do/while* loops

Module 12: Developing and Using Methods

- Describe the advantages of methods and define worker and calling methods
- Declare and invoke a method
- Compare object and static methods
- Use overloaded methods

Module 13: Implementing Encapsulation and Constructors

- Use encapsulation to protect data
- Create constructors to initialize objects

Module 14: Creating and Using Arrays

- Code one-dimensional arrays
- Set array values using length attribute and a loop
- Pass arguments to the main method for use in a program
- Create two-dimensional arrays

Module 15: Implementing Inheritance

- Define and test your use of inheritance
- Explain abstraction
- Explicitly identify class libraries used in your code