



## Java SE 8 Fundamentals

This Java SE 8 Fundamentals training introduces you to object-oriented programming using the Java language. Through hands-on exercises, you'll begin to build a baseline of knowledge to propel your career in development.

### Learn To:

- Use Java programming language constructs to create a Java technology application.
- Use decision and looping constructs and methods to dictate program flow.
- Understand basic object oriented concepts such as inheritance, encapsulation, and abstraction.
- Use and manipulate object references, and to write simple error handling code.
- Use the new SE 8 `java.time` and `java.time.format` packages to format and print the local date and time.
- Specify a data modification by passing a predicate lambda expression to the Collections class.

### Benefits to You

By enrolling in this course, you'll expand your knowledge of Java SE 8, while building your Java skill set. You'll build a solid basis in the Java programming language upon which to base continued work and training.

### Audience

- Administrator
- Developer
- Implementer/Manager
- Systems Administrator

### Objectives

- Write Java code that uses variables, arrays, conditional and loop constructs
- Manipulate primitive numeric data and string data using Java operators
- Create Java classes and use object references
- Access the fields and methods of an object
- Manipulate text data using the methods of the `String` and `StringBuilder` classes

- Use casting without losing precision or causing errors
- Declare, override, and invoke methods
- Access and create static fields and methods
- Use classes from the java.time and java.time.format packages to format and print the local date and time
- Encapsulate a class using access modifiers and overloaded constructors
- Define and implement a simple class hierarchy
- Demonstrate polymorphism by implementing a Java Interface
- Use a Predicate Lambda expression as the argument to a method
- Handle a checked exception in a Java application

## Topics

- What Is a Java Program?
  - Introduction to Computer Programs
  - Key Features of the Java Language
  - The Java Technology and Development Environment
  - Running/testing a Java program
- Creating a Java Main Class
  - Java Classes
  - The main Method
- Data In the Cart
  - Introducing variables
  - Working with Strings
  - Working with numbers
  - Manipulating numeric data
- Managing Multiple Items
  - Working with Conditions
  - Working with a List of Items
  - Processing a list of items
- Describing Objects and Classes
  - Working with objects and classes
  - Defining fields and methods
  - Declaring, Instantiating, and Initializing Objects
  - Working with Object References
  - Doing more with Arrays
  - Introducing the NetBeans IDE
  - Introducing the Soccer League Use Case
- Manipulating and Formatting the Data in Your Program
  - Using the String Class
  - Using the Java API Docs
  - Using the StringBuilder Class
  - More about primitive data types
  - The remaining numeric operators
  - Promoting and casting variables
- Creating and Using Methods
  - Using methods
  - Method arguments and return values
  - Static methods and variables
  - How Arguments are Passed to a Method
  - Overloading a method
- Using Encapsulation
  - Access Control
  - Encapsulation
  - Overloading constructors

- More on Conditionals
  - Relational and conditional operators
  - More ways to use if/else constructs
  - Using Switch Statements
  - Using the NetBeans Debugger
- More on Arrays and Loops
  - Working with Dates
  - Parsing the args Array
  - Two-dimensional Arrays
  - Alternate Looping Constructs
  - Nesting Loops
  - The ArrayList class
- Using Inheritance
  - Overview of inheritance
  - Working with subclasses and superclasses
  - Overriding methods in the superclass
  - Introducing polymorphism
  - Creating and extending abstract classes
- Using Interfaces
  - Polymorphism in the JDK foundation classes
  - Using Interfaces
  - Using the List Interface
  - Introducing Lambda expressions
- Handling Exceptions
  - Handling Exceptions: An overview
  - Propagation of exceptions
  - Catching and throwing exceptions
  - Handling multiple exceptions and errors