

# Java Deep Dive Training

## Table of Contents

### **Object-Oriented Programming with Java**

- Java Classesn
- Constructors and Garbage Collectionn
- Relationships Between Classesn
- Using thisn
- Visibilityn
- Packages and Importsn
- Overloading Methods and Constructorsn
- JARsn

### **Inheritance and Polymorphism in Java**

- Extending Classesn
- Using Derived Classesn
- Polymorphismn
- Overriding Methodsn
- Superclass Reference

### **Exception Handling**

- Reporting and Trapping Errorsn
- Exception Handlingn
- Throwing Exceptionsn
- Declaring Exceptions per Methodn
- Catching Exceptionsn
- The finally Blockn
- Catch-and-Releasen
- Chaining Exceptions

### **Using Classes Effectively**

- Class Loadingn
- Static Membersn
- Statics and Non-Statics
- Static Initializersn
- Static Importsn
- Prohibiting Inheritance -

## **Interfaces and Abstract Classes**

- Separating Interface and Implementation
- Defining Interfaces
- Implementing and Extending Interfaces
- Abstract Classes -

## **Collections**

- Dynamic Collections vs. Arrays
- Generics
- Using Generics
- The Collections API -

## **Generics**

- Using Generics
- Type Erasure
- Type Boundaries
- Wildcards
- Generic Methods
- Strengths and Weaknesses of Generics
- Legacy Code and Generics -

## **The Java Streams Model**

- Delegation-Based Stream Model
- InputStream and OutputStream
- Readers and Writers
- Java Serialization
- File Streams
- Sockets -

## **Threads**

- Java Thread Model
- Creating and Running Threads
- join and sleep
- Thread Synchronization
- wait and notify
- The Concurrency API –

## **Annotations**

- The Annotations Model
- Annotation Types and Annotations
- Built-In Annotations
- Annotations vs. Descriptors (XML)

## **Reflection**

- The Reflection API
- The Class
- Reading Type Information
- Dynamic Instantiation
- Dynamic Invocation –

## **Unit Testing with JUnit**

- The @Test Annotation
- The Assert Class Utility
- Test Runners
- Lifecycle Methods -