

# Java SE, Part 2 of 4: Inheritance, Interfaces, and Collections

page 1

**Meet the expert:** Gene Van Sant has over two decades of experience as a software/systems implementer and solutions integration architect. He has experience as a contractor and maintains instructor certifications from IBM and others. Gene balances his career between implementing and automating full blown real world solutions and teaching classes/workshops on emerging technologies and products. This allows Gene to gain valuable extensive real world experience while also staying ahead of the wave with cutting-edge expertise.

**Prerequisites:** This course is intended for programmers who are interested in adding Java to their skills. The students should be familiar with object-oriented principals and the concept of object-oriented programming.

**Runtime:** 08:49:28

**Course description:** This course introduces the Java programming language and how to develop Java applications using Eclipse. Students learn the syntax of the Java programming language, object-oriented programming using Java, exception handling, generics, collections, and file input/output (I/O) with a focus on many of the common programming Java commands like Arrays, and Inheritance. The course will conclude with Exceptions, Polymorphism, Collections, Generics and other useful Java classes.

## Course outline:

### Arrays

- Introduction
- Array Overview
- Arrays
- Declaring Arrays
- Populating Arrays
- Accessing Arrays
- Array Length
- Coding Tips:Arrays
- Demo:Arrays
- Array Referances
- Multidimensional Arrays
- Arrays of Arrays
- Demo:Arrays\_2D
- Copying Arrays
- For-Each Loop
- Variable Arguments
- Variable Arguments Example
- Demo:Arrays\_NewCalls
- Summary

### Inheritance

- Introduction
- Inheritance Is...
- Inheritance Examples
- Declaring Inheritance
- Inheritance Hierarchy
- Demo:Inheritance
- Access Modifiers Revisited

- Demo:Access Modifiers
- Summary

### Inherited Members

- Introduction
- Inherited Members
- Instances Of A Subclass
- Example of Inheritance
- Role In Reuse
- The Super Keyword
- Super Keyword Example
- Problems with Constructors
- Limiting Subclasses
- Demo:Final Keyword
- "set" Methods in Constructors
- The Object Class
- Demo:Inheritance
- Summary

### Overriding Methods

- Introduction
- Overriding Methods
- toString()
- toString() in Object
- Overriding toString()
- Demo:Overriding
- Memory review
- Comparing Objects
- Using == vs. equals(..)
- Demo:Primitives & References
- Summary

### Overriding Equals

- Introduction

- Overriding equals(..)
- Complex Comparisons
- equals(..) Example
- Hashcodes:Overview
- hashCode()
- Demo:Hashcodes
- Overriding hashCode()
- hashCode() Example
- @Override Annotation
- Demo:Equivalency
- Summary

### Exceptions

- Introduction
- What is an Exception
- Benefits
- Exception Processing:Overview
- The Exception Class
- How to Work With Exceptions
- Example Exception Handling
- try-catch-finally Statement
- Flow of Program Control
- Demo:Exception Processing
- Summary

### Exception Hierarchy

- Introduction
- Exception Hierarchy
- Checked Exceptions
- Unchecked Exceptions
- Coding Tlps - Exception Types

- Catching Multiple Exceptions
- Specifying Thrown Exceptions
- Rethrowing Exceptions
- Chaining Exceptions
- Creating your Own Exception
- Demo:Exceptions
- Demo:Create your Own Exception
- Assertions
- Assertion Example
- Demo:Assertions
- Summary

### Interfaces & Polymorphism

- Introduction
- Casting Objects
- The instanceof Operator
- Abstract Classes
- Abstract Class - An Example
- Demo:Abstract
- Interface
- Interface - An Example
- Comparable Interface
- Comparable Interface Reqs.
- Demo:Comparable Interfaces
- CodingTips
- Overview:Design Specs
- Polymorphism
- Conditions for Polymorphism
- Coding Tips - Polymorphism

(Continued on page 2)

# Java SE, Part 2 of 4: Inheritance, Interfaces, and Collections

page 2

- Covariant Return Types
- Overview:Interfaces
- Summary

## **Collections & Generics**

- Introduction
- What are Collections
- Arrays vs. Collections
- Main Collections Interfaces
- java.util.Collection
- Main Collection Methods
- Sets
- java.util.List
- java.util.Queue
- Generics:Overview
- Generics
- Generics and Collections
- Generic Collection Example
- Collections Implementations
- Demo:Collections
- Summary

## **Collections & Generics 2**

- Introduction
- Iteration on a Collection
- Iterator vs. For-Each Loop
- Demo:Iterator
- Maps
- java.util.Map
- java.util.SortedMap
- Collections Implementations
- Overview:Maps
- Demo:Maps
- Summary

## **Collections & Generics3**

- Introduction
- Collections & Primitive Types
- "Wrapper" Classes
- Autoboxing
- Demo:Autoboxing
- Enumerated Types
- Summary

## **JavaLogging & More**

- Introduction
- Java Logging API
- Control Flow of Logging
- Overview:LoggingInteraction
- Logging Levels
- Logging Handlers
- Loggers
- Logging Example
- Demo:Log Statements
- Logging Formatters&Log Manager
- Logging Configuration File
- Ex: Logging Configuration File

- Demo:Logging Configuration
- Summary

## **JavaLogging & More 2**

- Introduction
- Loggin Filters
- java.lang.StringBuilder
- java.util.StringTokenizer
- java.util.Arrays&Collections
- java.util.Random
- java.util.Date
- GregorianCalendar&Calendar
- Formatting
- Formatting Example
- Overview:Logging
- Demo:Utils
- Summary

## **Input & Output #1**

- Introduction
- Overview of Java Input/Output
- Streams
- Input Stream
- Output Stream
- "Chained" Streams
- The File Class
- Demo:File Object
- RandomAccessFile
- Reader and Writer
- Options for File IO
- Demo:Readers&Writers
- Summary

## **Input & Output #2**

- Introduction
- Buffers
- Channels
- Demo:Accessing File Channels
- Overview:Java RunTime
- Summary

## **Input & Output #3**

- Introduction
- Serialization
- Serializing Object State
- Overview:Composition Reference
- Avoid Serialization Problems
- serialVersionUID
- Demo:Serialization
- Summary