

Java Essentials (4 DAYS)

By Dr. Vishwanath Rao

Overview of Java

- History Of Java
- Benefits Of Java
- What Is Java?
- What's This "Virtual Machine"?
- Comparison to Other Languages
- Java Programs
- Basic Java Development Tools
- Java Editions
- Example – HelloWorld
- Java Classes
- Main Methods
- Statements
- Summary

Java Tools in Eclipse

- Eclipse Platform
- Eclipse Workspace
- Perspectives, Views & Editors
- Basic Operations with Eclipse Views and Perspectives
- The Java Perspective
- The Debug Perspective
- Navigator View
- Package Explorer
- Outline View
- Problems View
- Eclipse Preferences
- Build and Validation
- Code Completion, Templates and Snippets
- Searching
- Configure Compiler Class Path
- JRE Switching

Basic Object Concepts

- What Is An Object?
- State
- Behavior
- Encapsulation

- Encapsulation Examples
- Classes vs. Objects
- Inheritance
- Interfaces
- Polymorphism
- Benefits Of Objects
- Summary

Basic Java Syntax

- Declaring And Initializing Variables
- Keywords
- Coding Tips – Variables
- Primitive Data Types
- Logical - boolean
- Textual - char and String
- Integral - byte, short, int, long
- Floating Point - float and double
- Literal Values
- Strings
- Creating Strings
- White Space
- Comments
- Coding Tips - Comments
- Java Statements
- Coding Tips - Statements
- Scope of a Variable
- System.out/System.in
- Scanner Class
- Summary

Operations and Making Decisions

- Operator Categories
- Special Situations
- Binary Operators
- Integer Division
- Numeric Promotion
- Type Conversion Of Primitive Types
- Unary Operators
- Relational Operators
- Logical Operators
- "Short Circuited" Operators
- Bitwise Operators

- Bitwise Examples
- Shift Operators
- Overflow And Underflow
- Assignment Operators
- Ternary Operator
- Calculation Errors
- Operator Precedence
- Precedence Examples
- Combining Strings
- Coding Tips - Operators
- Control Flow Statements
- 'if' Statement
- 'if...else' Statement
- Nested Statements
- Coding Tips - if & if-else
- Summary

Using Classes and Objects

- Objects, Instances, And Classes
- What Are Classes?
- Working With Classes And Objects
- Instantiation
- Instance Methods
- Object References
- String Operations
- "Wrapper" Classes
- Autoboxing
- Summary

Writing Classes

- Why Define Your Own Classes?
- Encapsulation
- Elements Of A Class
- Defining Classes
- Coding Tips - Class Definitions
- Fields
- Defining Fields
- Coding Tips - Fields
- Methods
- Defining Methods
- Passing Parameters
- Overloading Methods

- Coding Tips - Methods
- Local Variables vs. Instance Variables
- Example - Defining a Class
- Example - Fields
- Example - Defining a Method
- Example - Calling a Method
- Summary

Controlling Code Access and Code Organization

- Controlling Access
 - Data Hiding
 - Encapsulation
 - JavaBeans
 - Packages
 - Naming Packages
 - Declaring Packages In Classes
 - Problems Solved With Packages
 - Package Access
 - Example - Access Modifiers
 - Import Statement
 - Using Classes From Packages
 - Coding Tips - Import Statements
 - Correlation To File Structure
 - Class Path
 - Java Core Packages
 - Java API Documentation
- Summary

Constructors and Class Members

- Constructors
 - Default Constructor
 - Multiple Constructors
 - Defining Constructors
 - Example - Calling Constructors
 - "Good" Constructors
 - 'this' Keyword
 - Using 'this' to Call a Constructor
 - Using 'this' to Set a Field
- Class Members
 - Examples Of Class Members
 - Comparison With Instance Members
 - Use Of Class Variables

- Static Class Methods
- Use Of Class Methods
- The Math Class
- Main Method And Command Line Arguments
- Declaring Constants
- Coding Tips - Class Members
- Useful Standard Class Members
- Initialization Blocks
- Static Initialization Blocks
- Summary

Advanced Control Structures

- 'switch' Statement
- Example - switch
- Switch "Fall Through"
- Using switch "Fall Through" for Multiple Options
- 'for' Loop
- Example - for
- 'while' Loop
- Example - while
- 'do...while' Loop
- Example - do while
- Break Statement
- Example - break
- Labeled Statements
- Example - Labeled break
- Continue Statement
- Example - continue
- Example - Labeled continue
- Coding Tips - Control Structures
- Summary

Arrays

- Arrays
- Declaring Arrays
- Populating Arrays
- Accessing Arrays
- Arrays of Objects
- Array Length
- Coding Tips - Arrays
- Array References
- Multidimensional Arrays

- Arrays Of Arrays
- Copying Arrays
- For-Each loop
- Variable Arguments
- Variable Arguments Example
- Summary

Inheritance

- Inheritance Is...
- Inheritance Examples
- Declaring Inheritance
- Inheritance Hierarchy
- Access Modifiers Revisited
- Inherited Members
- Instances Of A Subclass
- Example Of Inheritance
- Role In Reuse
- The super Keyword
- Example - super Keyword
- Problems with Constructors
- Limiting Subclasses
- Calling Methods in Constructors
- The Object Class
- Summary

Commonly Overridden Methods

- Overriding Methods
- @Override Annotation
- toString()
- toString() in Object
- Overriding toString()
- Comparing Objects
- Using == vs. equals(..)
- Overriding equals(..)
- Complex Comparisons
- equals(..) Example
- hashCode()
- Overriding hashCode()
- hashCode() Example
- Generating equals and hashCode
- Summary

Advanced Java Tools

- Refactoring
- Renaming Elements
- Moving a Class to a Different Package
- Extracting Code to a Method
- Other Source Code Refactoring
- Refactoring to Improve Type Hierarchy
- Generalizing a Variable
- Pull-up and Push-down

Exceptions

- What is an Exception
- Benefits
- The Exception Class
- How to Work With Exceptions
- Example Exception Handling
- The try-catch-finally Statement
- Flow of Program Control
- Exception Hierarchy
- Checked Exceptions
- Unchecked Exceptions
- Coding Tips - Exception Types
- Catching Multiple Exceptions
- Specifying Thrown Exceptions
- Rethrowing Exceptions
- Chaining Exceptions
- Creating your Own Exception
- Summary

Interfaces and Polymorphism

- Casting Objects
- The instanceof Operator
- Abstract Classes
- Abstract Class – An Example
- Interface
- Interface – An Example
- Comparable Interface
- Comparable Example
- Coding Tips - Superclass or Abstract Class/Interface?
- Coding Tips – Abstract Class or Interface
- Polymorphism
- Conditions for Polymorphism

- Coding Tips - Leveraging Polymorphism
- Covariant Return Types
- Covariant Return Types – An Example
- Summary

Collections and Generics

- What are Collections?
- Arrays vs. Collections
- Main Collections Interfaces
- `java.util.Collection`
- Main Collection Methods
- Sets
- `java.util.List`
- `java.util.Queue`
- Iteration on a Collection
- Iterator vs. For-Each Loop
- Maps
- `java.util.Map`
- Other Maps
- Collections Implementations
- Abstract Implementations
- Choosing a Collection Type
- Generics
- Generics and Collections
- Generic Collection Example
- Collections and Primitive Types
- Summary

Useful Java Classes

- Java Logging API
- Control Flow of Logging
- Logging Levels
- Loggers
- Logging Example
- Logging Handlers
- Logging Formatters & Log Manager
- Logging Configuration File
- Example Logging Configuration File
- Logging Filters
- `java.lang.StringBuilder`
- `java.util.StringTokenizer`
- `java.util.Arrays` & `java.util.Collections`

- java.util.Random
- java.util.Date
- GregorianCalendar & Calendar
- Formatting
- Formatting Example
- Summary

Input and Output

- Overview of Java Input/Output
- The File Class
- File Example
- Serialization
- Serializing Object State
- Avoiding Serialization Problems
- serialVersionUID
- Options for File Input/Output
- Streams
- Input Stream
- Output Stream
- "Chained" Streams
- RandomAccessFile
- Using Streams - Write Example
- Using Streams - Read Example
- Reader and Writer
- Using Readers and Writers - Write Example
- Using Readers and Writers - Read Example
- Using Readers and Writers - Scanner Read Example
- Buffers
- Channels
- Using Buffers and Channels - Write Example
- Using Buffers and Channels - Read Example
- Summary

Other Java Concepts

- Annotations
- Enumerated Types
- Enumerated Types – Example
- Assertions
- Assertions Example
- When to use Assertions
- Enabling Assertions
- JVM Storage Areas

- Java Heap Space
- Heap Size Limits
- Garbage Collection Basics
- Allocation Failure (AF)
- OutOfMemoryError
- Memory Leak
- Distributing Java Code with JARs