



JAVA Training Syllabus

JAVA

Declarations and Access Control

Class and Object

State (instance variables)

Behavior (methods)

Identifiers and Keywords

Inheritance

Interfaces

Finding Other Classes

Identifiers & JavaBeans

Legal Identifiers

Sun's Java Code Conventions

JavaBeans Naming Standards

Java Keywords

Methods

Variables

JavaBeans Standards

Declare Classes

Source File Declaration Rules

Class Declarations and Modifiers

Class Access

Default Access

Public Access

 Other (Nonaccess) Class Modifiers

 Final Classes

 Abstract Classes

Creating an Abstract Superclass and Concrete Subclass

Declare Interfaces

Declaring Interface Constants

Declare Class Members

Access Modifiers

Public Members

Private Members

Protected and Default Members

Protected Details

Default Details

Local Variables and Access Modifiers

Final Methods

Final Arguments

Abstract Methods

Synchronized Methods

Methods with Variable Argument Lists (var-args)

Constructor Declarations

Variable Declarations

Declaring Primitives and Primitive Ranges

Declaring Reference Variables

Instance Variables

Local (Automatic/Stack/Method) Variables

Array Declarations

Declaring an Array of Primitives

Declaring an Array of Object References

Final Variables

Transient Variables

Volatile Variables

Static Variables and Methods

Declaring Enums

Declaring Constructors, Methods, and Variables in an enum

Object Orientation

Encapsulation

Inheritance, Is-A, Has-A

IS-A and HAS-A Relationships

IS-A

HAS-A

Polymorphism

Overriding / Overloading

Overridden Methods

Invoking a Superclass Version of an Overridden Method

Examples of Legal and Illegal Method Overrides

Overloaded Methods

Legal Overloads

Invoking Overloaded Methods

Polymorphism in Overloaded and Overridden Methods

Reference Variable Casting

Implementing an Interface

Legal Return Types

Return Type Declarations

Return Types on Overloaded Methods

Overriding and Return Types, and Covariant Returns

Returning a Value

Constructors and Instantiation

Constructor Basics

Constructor Chaining

Rules for Constructors

Determine Whether a Default Constructor Will Be Created

What happens if the super constructor has arguments?

Overloaded Constructors

Statics

Static Variables and Methods

Accessing Static Methods and Variables

Coupling and Cohesion

Coupling

Cohesion

Assignments

Stack and Heap

Literals, Assignments, and Variables

Literal Values for All Primitive Types

Integer Literals

Decimal Literals

Octal Literals

Hexadecimal Literals

Floating-Point Literals

Boolean Literals

Character Literals

Literal Values for Strings

Assignment Operators

Primitive Assignments

Primitive Casting

Casting Primitives

Assigning Floating-Point Numbers

Assigning a Literal That Is Too Large for the Variable

Assigning One Primitive Variable to Another Primitive Variable

Reference Variable Assignments

Variable Scope

Using a Variable or Array Element That Is Uninitialized and Unassigned

Primitive and Object Type Instance Variables

Primitive Instance Variables

Object Reference Instance Variables

Array Instance Variables

Local (Stack, Automatic) Primitives and Objects

Local Primitives

Local Object References

Local Arrays

Assigning One Reference Variable to Another

Passing Variables into Methods

Passing Object Reference Variables

Does Java Use Pass-By-Value Semantics?

Passing Primitive Variables

Array Declaration, Construction, and Initialization

Declaring an Array

Constructing an Array

Constructing One-Dimensional Arrays

Constructing Multidimensional Arrays

Initializing an Array

Initializing Elements in a Loop

Declaring, Constructing, and Initializing on One Line

Constructing and Initializing an Anonymous Array

Legal Array Element Assignments

Arrays of Primitives

Arrays of Object References

Array Reference Assignments for One-Dimensional Arrays

Array Reference Assignments for Multidimensional Arrays

Initialization Blocks

Using Wrapper Classes and Boxing

An Overview of the Wrapper Classes

Creating Wrapper Objects

The Wrapper Constructors

The `valueOf()` Methods

Using Wrapper Conversion Utilities

`xxxValue()`

`parseXxx()` and `valueOf()`

`toXxxString()` (Binary, Hexadecimal, Octal)`toString()`

Autoboxing

Boxing, ==, and equals()

Where Boxing Can Be Used

Overloading

Overloading Made Hard – Method Matching

Overloading with Boxing and Var-args

Widening Reference Variables

Overloading When Combining Widening and Boxing

Overloading in Combination with Var-args

Garbage Collection

Overview of Memory Management and Garbage Collection

Overview of Java's Garbage Collector

Writing Code That Explicitly Makes Objects Eligible for
Collection

Reassigning a Reference Variable

Isolating a Reference

Nulling a Reference

Forcing Garbage Collection

Cleaning Up Before Garbage Collection – the finalize()
Method

Tricky Little finalize() Gotcha's

Java Operators

Assignment Operators

Compound Assignment Operators

Relational Operators

Equality Operators

Equality for Primitives

Equality for Reference Variables

Equality for Enums

instanceof Comparison

instanceof Compiler Error

Arithmetic Operators

The Remainder (%) Operator

String Concatenation Operator

Increment and Decrement Operators

Conditional Operator

Logical Operators

Flow Control, Exceptions, and Assertions

if and switch Statements

if-else Branching

Legal Expressions for if Statements

switch Statements

Legal Expressions for switch and case

Break and Fall-Through in switch Blocks

The Default Case

Creating a switch-case Statement

Loops and Iterators

Using while Loops

Using do Loops

Using for Loops

The Basic for Loop

The Basic for Loop: Declaration and Initialization

Basic for Loop: Conditional (boolean) Expression

Basic for Loop: Iteration Expression

Basic for Loop: for Loop Issues

The Enhanced for Loop (for Arrays)

Using break and continue

Unlabeled Statements

Labeled Statements

Creating a Labeled while Loop

Handling Exceptions

Catching an Exception Using try and catch

Using finally

Propagating Uncaught Exceptions

Propagating and Catching an Exception

Defining Exceptions

Exception Hierarchy
Handling an Entire Class Hierarchy of Exceptions
Exception Matching
Exception Declaration and the Public Interface
Rethrowing the Same Exception
Creating an Exception
Common Exceptions and Errors
Where Exceptions Come From
JV M Thrown Exceptions
Programmatically thrown exceptions
Working with the Assertion Mechanism
Assertions Overview
Assertion Expression Rules
Enabling Assertions
Identifier vs. Keyword
Enabling Assertions at Runtime
Disabling Assertions at Runtime
Selective Enabling and Disabling

Flow Control, Exceptions, and Assertions

String, StringBuilder, and StringBuffer

The String Class

Strings Are Immutable Objects

Important Facts About Strings and Memory

Creating New Strings

Important Methods in the String Class

```
public char charAt(int index)
```

```
public String concat(String s)
```

```
public boolean equalsIgnoreCase(String s)
```

```
public int length()
```

```
public String replace(char old, char new)public String
```

```
substring(int begin)
```

```
public String substring(int begin, int end)
```

```
public String toLowerCase()
```

```
public String toString()
```

```
public String toUpperCase()
```

```
public String trim()
```

The StringBuffer and StringBuilder Classes

StringBuffer vs. StringBuilder

Using StringBuilder and StringBuffer

Important Methods in the StringBuffer and StringBuilder

Classes

```
public synchronized StringBuffer append(String s)
```

```
public StringBuilder delete(int start, int end)
```

```
public StringBuilder insert(int offset, String s)
```

```
public synchronized StringBuffer reverse()
```

public String toString()

File Navigation and I/O

The Stream Classes

The Byte Streams

InputStream

OutputStream

File

FileReader

BufferedReader

FileWriter

BufferedWriter

PrintWriter

Creating Files Using Class File

First execution

Second execution

Using FileWriter and FileReader

Combining I/O classes

Working with Files and Directories

Serialization

Working with ObjectOutputStream and ObjectInputStream

Object Graphs

Using writeObject and readObject

How Inheritance Affects Serialization

Serialization Is Not for Statics

Dates, Numbers, and Currency

Working with Dates, Numbers, and Currencies

Orchestrating Date- and Number-Related Classes

The Date Class

The Calendar Class

The DateFormat Class

The Locale Class

The NumberFormat Class

Parsing, Tokenizing, and Formatting

A Search Tutorial

Simple Searches

Searches Using Metacharacters

Searches Using Quantifiers

The Predefined Dot

Greedy Quantifiers

When Metacharacters and Strings Collide

Locating Data via Pattern Matching

Searching Using the Scanner Class

Tokenizing

Tokens and Delimiters

Tokenizing with `String.split()`

Tokenizing with Scanner

Formatting with printf() and format()

Generics and Collections

Overriding hashCode() and equals()

The toString() Method

Overriding equals()

Implementing an equals() Method

The equals() Contract

Overriding hashCode()

Understanding Hashcodes

Implementing hashCode()

The hashCode() Contract

Collections

Key Interfaces and Classes of the Collections Framework

List Interface

ArrayList

Vector

LinkedList

Set Interface

HashSet

LinkedHashSet

TreeSet

Map Interface

HashMap

Hashtable

LinkedHashMap

TreeMap

Queue Interface

PriorityQueue

Using the Collections Framework

ArrayList Basics

Autoboxing with Collections

Sorting Collections and Arrays

Sorting Collections

The Comparable Interface

Sorting with Comparator

Sorting with the Arrays Class

Searching Arrays and Collections

Converting Arrays to Lists to Arrays

Using Lists

Using Sets

Using Maps

Using the PriorityQueue Class

Method Overview for Arrays and Collections

Method Overview for List, Set, Map, and Queue

Generic Types

The Legacy Way to Do Collections

Generics and Legacy Code

Mixing Generic and Non-generic Collections

Polymorphism and Generics

Generic Methods

Generic Declarations

Making Your Own Generic Class

Creating Generic Methods

Inner Classes

Inner Classes

Coding a "Regular" Inner Class

Instantiating an Inner Class

Instantiating an Inner Class from Within the Outer Class

Creating an Inner Class Object from Outside the Outer Class

Instance Code

Referencing the Inner or Outer Instance from Within the

Inner Class

Member Modifiers Applied to Inner Classes

Method-Local Inner Classes

Anonymous Inner Classes

Plain-Old Anonymous Inner Classes, Flavor One

Plain-Old Anonymous Inner Classes, Flavor Two

Argument-Defined Anonymous Inner Classes

Static Nested Classes

Instantiating and Using Static Nested Classes

Threads

Defining, Instantiating, and Starting Threads

Making a Thread

Defining a Thread

Extending `java.lang.Thread`

Implementing `java.lang.Runnable`

Instantiating a Thread

Starting a Thread

Starting and Running Multiple Threads

The Thread Scheduler

Methods from the `java.lang.Thread` Class

Methods from the `java.lang.Object` Class

Thread States and Transitions

Thread States

Runnable

Running

Preventing Thread Execution

Sleeping

Creating a Thread and Putting It to Sleep

Thread Priorities and `yield()`

Setting a Thread's Priority

The `yield()` Method

The `join()` Method

Synchronizing Code

Preventing the Account Overdraw

Synchronization and Locks

So What About Static Methods? Can They Be Synchronized?

Synchronizing a Block of Code

What Happens If a thread Can't Get the Lock?

So When Do I Need To Synchronize?

Thread-Safe Classes

Thread Deadlock

Thread Interaction

Using `notifyAll()` When Many Threads May Be Waiting

Using `wait()` in a Loop

The Applet Class

Applet Basics

Applet Architecture

Applet Initialization and Termination

Requesting Repainting

The HTML APPLET Tag

getDocumentBase() and getCodeBase()

AppletContext and showDocument()

Event Handling

The Delegation Event Model

Event Sources

Event Listeners

Event Classes

The ActionEvent Class

The AdjustmentEvent Class

The ComponentEvent Class

The ContainerEvent and ContainerEvent Class

The FocusEvent Class

The InputEvent Class

The ItemEvent Class

The KeyEvent Class

The MouseEvent Class

The TextEvent Class

The WindowEvent Class

Sources of Events

Event Listener Interfaces

The ActionListener Interface

The ItemListener Interface

The KeyListener Interface

The MouseListener Interface
The MouseMotionListener Interface
The TextListener Interface
The WindowListener Interface
Handling Mouse Events
Handling Keyboard Events
A dapter Classes

AWT: Working with Windows, Graphics, and Text

A WT Classes
W indow Fundamentals
Component
Container
Panel
Window
Frame
Canvas
W orking with Frame Windows
Setting the Window's Dimension
Hiding and Showing a Window
C reating a Frame Window in an Applet
Creating a Windowed Program

Working with Graphics

Working with Color

Color Methods

Setting the Current Graphics Color

Working with Fonts

Using AWT Controls, Layout Managers, and Menus

Control Fundamentals

Adding and Removing Controls

Responding to Controls

Understanding Layout Managers

Menu Bars and Menus

Dialog Boxes

FileDialog

Handling Events by Extending AWT Comp

A Tour of Swing

JApplet

JFrame

Icons and Labels

Exploring Swing

LookAndFeel For Window

Creating JAR File and its execution

Remote Method Invocation (RMI) And Networking

What is RMI?

RMI Server

RMI Client

Generating Stubs

RMI Registry

Running RMI Programs

Networking Details

Socket Overview

Reserved Sockets

Java and the Net

The Networking Classes and Interfaces

InetAddress

Factory Methods

Instance Methods

TCP/IP Client Sockets

URLConnection

TCP/IP Server Sockets

Datagrams

DatagramPacket

Datagram Server and Client

Databases

Database overview

Jdbc and odbc

Type of JDBC

The database connection

Driver loading

Retrieving value from database

Insert value into database

Difference between Statement and PreparedStatement

Connection closing