Introduction to Java

- Cornerstones of the Java Platform
- Java Advantages
- The Java Programming Language
- The Java Virtual Machine (JVM)
- Core Java Libraries
- Extension Libraries

Java Syntax Fundamentals

- Comments
- Identifiers
- Reserved Words
- Classes
- Statements and Blocks
- Variables, Constants, Literals
- Scope of Variables
- Methods
- Method Overloading
- static Members
- Static Import (Java SE 5+)
- Naming Conventions

Flow of Control

- `if/else` Statement
- Combining `ifs`
- `while` and `do/while` Loops
- `for` Loop and Loop Counters
- `break` and `continue`
- Break to Labeled Loops
- `switch` Statement
- `return` Statement
- Exit Status

Strings

- String Manipulation
- `StringBuffer` and `StringBuilder`
- Simple Number/String Conversion

Developing Java Classes

- Object-Oriented (OO) Concepts
- Methods, Member Variables
- Accessing Members
- Tight Encapsulation
- Access Control Modifiers
- Constructors and Finalizer
- Using `this`
- Class Variables - Static Members and Static Blocks
- Instance Variables
- Local Variables
- Variables and Initialization
- Inner Classes

- Anonymous Classes
- JavaBeans
- Driver Classes

Type Safety

- Annotations (Java SE 5+)
- Java SE Built-In Annotations
- Defining New Annotations
- Enumerated Types (Java SE 5+)
- Constants and Constrained Values
- Defining and Declaring `enums`
- `enum` Values
- `enums` and `switch` Statements
- `values()` and `valueOf()`
- Generic Classes (Java SE 5+)
- Generic Type Parameters
- Using Type Parameters `inClass`, Variable and Method Declarations
- Using a Generic Class
- Bounded Type Parameters

Exceptions and Exception Handling

- The Throwable Hierarchy: Error, `RuntimeException` and Checked Exception
- Methods that Throw Exceptions
- Handling Exceptions with `try-catch-finally` Blocks
- Application-Defined Exceptions
- Throwing an Exception
- Assertions (Java 1.4+)
- Enabling Assertions at Run-Time

Network Programming

- The `java.net` Package
- IP Addresses and Port Numbers
- Client/Server Socket Programming
- `URL` and `URLConnection` Classes
- Communicating with Web Servers
- HTTP `GET` and `POST` Operations
- Posting to a Server-Side Program

Java Database Connectivity

- The `java.sql` Package
- JDBC Architecture and Drivers
- SQL Exceptions
- `DriverManager`, `Connection`, `Statement` and `ResultSet` Interfaces
- Examining Database MetaData
- Basic Query and Update
- Improving Performance with `PreparedStatement` and `CallableStatement` Interfaces
- JDBC Transaction Management

JavaServer Pages (JSPs)

- JSP Lifecycle
- Elements of a JSP

- Directives, Declarative, Scriptlets
- Writing a JSP
- Objects Available in a JSP
- Repeated content in JSPs
- Translation-Time and Request-Time Includes
- Using JavaBeans in a JSP
- Session Management
- Mixing JSPs and Servlets
- Installing and Using Tag Libraries
- The JSP `taglib` Directive
- The Tag Library Descriptor

Developing Software Using Java

- Applications, Applets, Web Components
- Java SE, Java EE, Java ME
- Installing the JDK
- Compiling and Running Java from the Command Line
- The `main()` Method
- `package` and `import` Statements
- JAR Files
- Class Loading and `CLASSPATH`
- Online API Documentation
- JDK Tools
- Java Integrated Development Environments (IDEs)

Data Types and Operators

- Primitive Types
- Boolean, Integer, Floating-Point and Character Types
- Unicode Characters and Strings
- Type Conversion and Casting
- Expressions and Operators
- Arithmetic Operators
- Increment/Decrement Operators
- Division and Remainder Operators
- Assignment Operators
- Relational Comparison and Logical Operators
- Conditional Operator
- Bitwise Operators
- Order of Evaluation
- Operator Precedence and Associativity

Using Java Classes and Objects

- Classes as Data Types
- Objects and References
- Memory in the JVM
- Object Initialization
- Objects as Arguments to Methods
- Objects as Return Values
- Garbage Collection
- Primitive Wrapper Classes - Integer, Double, etc.
- Autoboxing and Unboxing (Java SE 5+)

Arrays

- Declaring and Allocating Arrays
- Multi-Dimensional Array
- Array Literals
- The `java.util.Arrays` Class
- Command-Line Arguments
- Enhanced for Loop (Java SE 5+)
- Arrays as Method Arguments
- Variable-Length Arglists (varargs) (Java SE 5+)
- Autoboxing and varargs

Inheritance

- Extending Java Classes
- Accessing Superclass Constructors and Members
- Overriding Methods
- Abstract CLasses and Methods
- Polymorphism
- Overriding Methods of `java.lang.Object`
- `equals()`, `toString()`, `hashCode()`
- Final Classes and Methods
- Multiple Inheritance
- Interfaces
- Casting Object References
- Documenting Classes with the `javadoc` Utility
- Unit Testing

The Collections Framework

- The `java.util` Package
- Container Objects
- Arrays as Containers
- Legacy Container Classes - Vector, Hashtable, Enumeration
- Legacy Container Generic Forms (Java SE 5+)
- Collections Interfaces - `Collection<E>`, `List<E>`, `Set<E>`, `SortedSet<E>`
- Map Interfaces - `Map<K,V>`
- Coding to the Interface
- `List<E>`, `Set<E>`, `Queue<E>` and `Map<K,V>` implementations
- Iterating Collections with the `Iterator<E>` Interface
- Collections and the Enhanced for Loop
- Choosing the Correct Implementation and Interface
- The `java.util.Collections` Utility Class
- Sorting Using the Comparable Interface

Basic Input and Output (I/O)

- The `java.io` Package
- Using Stream Classes
- Combining Streams
- `flush()` and `close()`
- Console Input and Output
- Navigating the File System
- File Streams
- Character File Input and Output
- Reader and Writer Interfaces
- `BufferedReader` and `BufferedWriter`
- Binary File I/O - `DataOutputStream` and `DataInputStream`

- Object Streams - `ObjectInputStream` and `ObjectOutputStream`
- Serialization and Versioning
- Random Access Files
- Formatted Input and Output
- Formatter (Java SE 5+)
- Format specifiers, `printf()` and `format()`
- `java.text` Classes for Formatting Dates, Numbers, Currencies
- Input with Scanner (Java SE 5+)

Threads

- Life and States of a Thread
- Creating and Starting a Thread
- `java.lang.Runnable` and `java.lang.Thread`
- Stopping a Thread
- Inter-Thread Communication
- Thread-Safe Access to Shared Objects and Variables
- Synchronized Code
- Sleeping
- Interrupting a Blocked Thread
- `wait()`, `notify()`, `notifyAll()` Example
- Thread Scheduling
- Thread Groups

Java Web Applications

- Java Enterprise Edition
- Java EE Application Servers
- Web Application Directory and WAR files
- Deploying a Web Application - The `web.xml` File
- Servlet Architecture
- The `javax.servlet` Package
- Servlet Classes and Interfaces
- Writing a Servlet
- `HttpServletRequest` and `HttpServletResponse`
- Handling HTML Forms
- Retrieving Request Parameters