



## **Sky Info technologies**

B-77, Sector-64, Distt. Gautam Budh Nagar (U.P.) 201307  
Tel. : +91-120-4242224, 4242223, Fax : +91-120-4242223  
Website : www.skyinfotechnologies.com

### **JAVA CURRICULUM**

**DURATION : 140 Hours**

#### **J2SE (JAVA 2 STANDARD EDITION)**

- **INTRODUCTION TO JAVA**
  - Understanding Requirement: Why Java
  - Why Java is important to the Internet
  
- **INTRODUCTION TO JAVA VIRTUAL MACHINE**
  - Java Virtual Machine Architecture
  - Class loading process by Classloaders
  - Role of Just in Time Compiler (JIT)
  - Execution Engine
  
- **PACKAGES AND INTERFACES**
  - Defining a Package
  - Understanding CLASSPATH
  - Access Protection
  - Importing Packages
  - Defining and implementing an Interface
  - Abstract classes Vs Interfaces
  - Adapter Classes
  - Anonymous Classes
  
- **AN OVERVIEW OF JAVA AND BUZZWORDS**
  - Data Types, Variables and Arrays
  - Operators
  - Control statements
  - Object oriented Paradigms
  - Abstraction
  - The Three OOP Principles:  
(Encapsulation, Inheritance and Polymorphism)
  
- **EXCEPTION HANDLING**
  - Fundamentals of Exception handling
  - Types of Exceptions
  - Learning exception handlers
  - Try and catch
  - Multiple catch clauses
  - Nested try statements
  - Throw, throws and finally
  - Creating custom exceptions
  
- **JAVA CLASSES AND OOP IMPLEMENTATION**
  - Class Fundamentals
  - Command Line Arguments
  - Learning static initializer

- Declaration of Objects
- Instance Variable Hiding
- Overloading and Overriding of Methods
- Understanding of Access Controls:
- Private, public and protected
- Learning Nested and Inner Classes
- Dynamic method Dispatching
- Using Abstract Classes
- Using final to prevent Overriding & Inheritance
- Garbage Collection
- **STRING HANDLING**
  - Learning String Operations
  - Learning Character Extraction
  - Learning String Comparison
  - Understanding of StringBuffer Class
- **New in JDK 5**
  - Generics
  - Annotations
  - Vargs
  - static-import
  - for-each
- **MULTITHREADED PROGRAMMING**
  - The Java Thread Model
  - Creating a Thread: Extending Thread,
  - Implementing Runnable
  - Creating Multiple Threads and Context
  - Switching
  - Synchronization: Methods and Statement
  - Inter-thread Communication
- **REFLECTION API**
  - Determining the class of an Object
  - Getting Information about Class's modifiers, fields, methods, constructors and super classes
  - Finding out constant and method declaration belong to an interface
  - Creating an instance of a class whose name is not known until runtime
  - Getting and setting value of an object's field if field name is unknown until runtime
  - Invoking a method on an object if the method is unknown until runtime
  - Creating a new array whose size and component type are not known until runtime.

## **ADVANCE JAVA & J2EE (JAVA 2 ENTERPRISE EDITION)**

- **COLLECTION FRAMEWORK**
  - The Collection Interfaces (List, Set, SortedSet)
  - The Collection Classes
  - (ArrayList, LinkedList, HashSet, TreeSet)
  - Accessing a Collection via an Iterator
  - Working with Maps
  - Working with Comparators
  - The Collection Algorithms
  - The Legacy Classes and Interfaces
  - (Enumeration, Vector, Stack, Dictionary, Hashtable)
  - Date and Time Handling

- **SYSTEM PROPERTIES & INTERNATIONALIZATION**
  - Usage of Property file
  - Define the Locale
  - ResourceBundle
  - Fetching text from ResourceBundle
  
- **REMOTE METHOD INVOCATION (RMI)**
  - Distributed Applications
  - RMI Architecture
  - Implementation
  - Call-back Mechanism
  
- **DATABASE PROGRAMMING USING JDBC**
  - JDBC Drivers
  - Statements
  - Metadata
  - Scrollable & Updatable ResultSet
  - Batch Updates
  - Data Source & Connection Pooling
  - Rowsets
  
- **INTRODUCTION TO J2EE ARCHITECTURE**
- **TIER ARCHITECTURE**
  - Single Tier
  - Two Tier
  - Three Tier
  - N Tier
  
- **J2EE COMPONENTS**
  - Web Components
  - Business Components
  
- **J2EE CONTAINERS**
  - Container Types
  - Container Services
  
- **J2EE SERVICES**
  - Java Naming and DirectoryInterface
  - Java Transaction Service
  - Java Messaging Service
  - Java Authentication & Authorization Service
  
- **INTRODUCTION TO UML**
  - Use Cases
  - Diagrams
  
- **INTRODUCTION TO XML**
  - Document Type Definition (DTD)
  - Schemas
  
- **JAVA SERVLET**
  - Introduction to Web Programming
  - Advantages of Servlets

- Servlet Lifecycle
- Request Dispatching
- Session Tracking
- Event Listener
- Filters
  
- **JAVA SERVER PAGES (JSP) & JSTL**
  - JSP Architecture
  - JSP Elements
  - JSP Objects
  - Custom Tags
  - Using tags of JSTL
  
- **ENTERPRISE JAVA BEANS (EJB)**
  - INTRODUCTION
  - ARCHITECTURE
  - TYPES OF EJB
  - SESSION BEANS
    - Introduction
    - State Management
    - Life Cycle
    - Types
      - Stateless
      - Statefull
  - ENTITY BEANS
    - Introduction
    - Persistence
    - Life Cycle
    - Types
      - Container Managed Persistent (CMP) 2.0
        - Introduction
        - Primary Key Class
      - EJB Methods
      - EJB QL
        - Syntax
        - Clauses
    - Bean Managed Persistent (BMP)
      - Introduction
      - Implementation
    - EJB Relationships
  - MESSAGE DRIVEN BEANS & JMS
    - Messaging Overview
    - Messaging Models
      - Point to Point Model
      - Topic Subscriber Model
    - JMS Implementation
    - Life Cycle
  
- **J2EE DESIGN PATTERN**
  - Why Design Patterns?
  - Front Controller
  - Composite View
  - Session Façade
  - Service Locator
  - Data Access Object
  - Value Object

- **JAVA MAIL**
  - Email System & Protocols
  - Architecture
  - Sending mails
  - Receiving mails
  - Handling attachments
  - Replying & Forwarding
  
- **PACKAGING & DEPLOYMENT USING ANT**
- **INTRODUCTION TO WEB SERVICES**
  - A Conceptual overview of Web Services
  - Web Services Requirements
  - SOAP
  - WSDL
  - UDDI
  
- **UTILITIES**
  - Eclipse 3.1
- **WEB SERVER**
  - Apache Tomcat
- **APPLICATION SERVER.**
  - Bea's Weblogic
  - Sun Application Server
  - Tomcat Server
  
- **TRANSACTIONS**
  - Introduction
  - ACID Properties
  - Transaction Requirements
  - Container managed transactions
  - Bean Managed Transactions
  - Distributed Transactions
  - Two Phase Commit

Live project which is based on Java Technology with the help of J2SE for representing the core java concept and J2EE which represents the evolution of Sun Microsystems's server side development is an essential part of developing and nurturing the skills for a **FRESHER** because it standardizes the development and deployment of the kind of secure, portable, reliable and scalable applications required by the World Wide Web Development.

- **LIVE PROJECT**
  - What is Live Project?
  - Project Phases
  - What is Project Life Cycle?
  - Software Development Cycle
  - What is Project Team?
  - What does Project Team do?
  - Roles and Responsibilities

