

INFOSOFT IT SOLUTIONS

Training | Projects | Placements

Revathi Apartments, Ameerpet, 1st Floor, Opposite Annapurna Block, Infosoft It solutions,
Software Training & Development Institute, +91 - 9059683947 | +91 - 9182540872

ENOVIA PLM Fundamentals

Overview of SAP PLM with the key areas

- Overview of SAP PLM
- Life-Cycle Data Management
- Life-Cycle Collaboration

Document management System (DMS)

- Use of the document info record
- Object links to other objects
- Use of originals
- Original storage in the secure area
- Documents Distribution
- Demo on Easy DMS
- Redlining

DMS Configuration

- Operators
- Conditionals
- Loops
- Operator Precedence
- Incrementing and Decrementing (++ and --)

- Unary NOT (~ And !)
- Multiplication and Division (* and /)
- Modulus (%)
- Addition and Subtraction (+ and -)
- Shift Operators (>>, >>>, and <<)
- Relational Operators (>, >=, <, <=, ==, and !=)
- Bitwise and Bitwise Logical AND, XOR, and OR (&, ^, and /)
- Logical (&& and ||)
- The if-then-else Operator
- Assignment Operators (= and [operator]=)
- Using the Math
- Changes in the Math Class
- Class StrictMath
- Comparing Strings
- The if Statement
- The else Statement
- Nested if
- The if-else Ladders
- The switch Statement
- Using Strings in switch Statement
- The while Loop
- The do-while Loop
- The for Loop
- The for-each Loop
- Supporting for-each in Your Own Class
- A (Poor) Solution
- Significance of for-
- Nested Loops

- Using the break Statement
- Using the continue Statement
- Using the return Statement
- Summary

Class, Object, Packages and Access Specifiers

- The Control Overview of a Class
- Working with Objects
- Working with Methods
- Defining Default Methods
- Working with Constructors
- Using Default Constructor
- Using Parameterized Constructors
- Exploring Packages
- Studying the Types of Packages
- Importing Packages
- Using Access Specifiers
- Working with Streams API
- Stream API Overview
- Collection and Stream
- Commonly Used Functional Interfaces in Stream
- Java.util.Optional
- Aggregate Operations
- Working with Time API

Implementing Object-Oriented Programming in Java

- Understanding Encapsulation
- Understanding Abstraction
- Understanding Inheritance

- Understanding the final Keyword
- Preventing Inheritance
- Declaring Constant
- Preventing Method Overriding
- Implementing Interfaces
- Working with Lambda Expressions
- Method References
- Using Lambda Expressions
- Implementing Abstract Classes and Methods
- Difference between Abstract Classes and Interfaces
- Implementing Polymorphism
- Understanding the Static Polymorphism
- Understanding the Dynamic Polymorphism
- Summary

Working with Streams, Files and I/O Handling

- Streams, Readers and Writers
- Essentials in NIO
- Buffers
- Channels
- Charsets and Selectors
- Enhancements in NIO with Java 8
- The Path Interface
- The Files Class
- The Paths Class
- The File Attribute Interfaces
- The FileSystem Class
- The FileSystems Class

- The FileStore Class
- Prospects of NIO
- Working with Streams
- The InputStream Class
- The OutputStream Class
- The ByteArrayInputStream Class
- The ByteArrayOutputStream Class
- The BufferedInputStream Class
- The BufferedOutputStream Class
- The FileInputStream Class
- The FileOutputStream Class
- Working with the Reader Class
- Working with the Writer Class
- Accepting Input from the Keyboard with the InputStreamReader Class
- Working with the OutputStreamWriter Class
- Working with Files
- Using the File Class
- Using the FileReader Class
- Using the FileWriter Class
- Working with the RandomAccessFile Class
- Working with Character Arrays
- Using the CharArrayReader Class
- Using the CharArrayWriter Class
- Working with Buffers
- Using the BufferedReader Class
- Using the BufferedWriter Class
- Working with the PushbackReader Class

- Working with the PrintWriter Class
- Working with the StreamTokenizer Class
- Implementing the Serializable Interface
- Working with the Console Class
- Working with the Clipboard
- Working with the Printer
- Printing with the Formatter Class
- Using the System.out.printf() Method
- Using the String.format() Method
- Formatting Dates Using the String.format() Method
- Using the Java.util.Formatter Class
- Scanning Input with the Scanner class
- Summary

Implementing Exception Handling

- Overview of Exceptions
- Exception Handling Techniques
- Rethrowing Caught Exception with Improved Type Checking
- Built-in Exceptions
- User-Defined Exceptions
- Summary

Working with Multiple Threads

- Using Threads in Java
- Life Cycle of a Thread
- Synchronization of Threads
- Multithreaded Custom Class Loader
- Getting the Main Thread
- Naming a Thread

- Pausing a Thread
- Creating a Thread with the Runnable Interface
- Creating a Thread with the Thread Class
- Creating Multiple Threads
- Joining Threads
- Checking if a Thread Is Alive
- Setting Thread Priority and Stopping Threads
- Synchronizing
- Communicating between Threads
- Suspending and Resuming Threads
- Creating Graphics Animation with Threads
- Eliminating Flicker in Graphics Animation Created Using Threads
- Suspending and Resuming Graphics Animation
- Using Double Buffering
- Simplifying Producer-Consumer with the Queue Interface
- Implementing Concurrent Programming
- Simplifying Servers Using the Concurrency Utilities
- Knowing Various Concurrency Utilities
- Learning about the Java.util.concurrent Package
- Learning about the Java.util.concurrent.locks Package
- Learning about the Java.util.concurrent.atomic Package
- Summary

Working with Collections Framework

- The Collection Interfaces
- The Collection Classes
- The Map Interfaces
- The Map Classes

- Collections Framework Enhancements in Java SE 8
- Using the Collection Interface
- The Queue Interface
- The List Interface
- The Set Interface
- The SortedSet Interface
- Using the Collection Classes
- Using the Comparator Interface
- Using the Iterator Interface
- Using the ListIterator Interface
- Using the AbstractMap Class
- Using the HashMap Class
- Using the TreeMap Class
- Using the Arrays Class
- Learning the Fundamentals of Enumerations
- The Legacy Classes and Interfaces
- Using the Aggregate Operations
- Using the Java.util.function Package
- Summary

Creating Packages, Interfaces, JAR Files and Annotations

- Packages and Interfaces
- JAR Files
- The Java API Package
- The Java.lang Package
- Basics of Annotation
- Other Built-In Annotations
- Creating a Package

- Creating Packages that have Subpackages
- Creating an Interface
- Implementing an Interface
- Extending an Interface
- Using Interfaces for Callbacks
- Performing Operations on a JAR File
- Marker Annotations
- Single Member Annotations
- Summary

Working with Java Beans

- What is Java Bean?
- Advantages of Java Bean
- Introspection
- Persistence
- Customizers
- Understanding Java Beans
- Designing Programs Using Java Beans
- Creating Applets that Use Java Beans
- Creating a Java Bean
- Creating a Bean Manifest File
- Creating a Bean JAR File
- Creating a New Bean
- Adding Controls to Beans
- Giving a Bean Properties
- Design Patterns for Properties
- Using Simple Properties
- Designing Patterns for Events

- Learning Methods and Design Patterns
- Creating Bound Properties
- Giving a Bean Methods
- Giving a Bean an Icon
- Creating a BeanInfo Class
- Setting Bound and Constrained Properties
- Implementing Persistence
- Using the Java Beans API
- Learning the Basics of an Event
- Using the Java Beans Conventions
- Using the Remote Notification and Distributed Notification
- Using Beans with JSP
- Summary

Networking and Security with Java

- Basics of Networking
- Sockets in Java
- Client-Server Networking
- Proxy Servers
- Internet Addressing
- Domain Name Service
- Inet4Addresses and Inet6Addresses
- The URL Class
- The URI Class
- URI Syntax and Components
- TCP/IP and Datagram
- Blackboard Assignment Retrieval Transaction
- Understanding Networking Interfaces and Classes in the Java.net Package

- Understanding the InetAddresses
- Caching InetAddress
- Creating and Using Sockets
- Creating TCP Clients and Servers
- Understanding the Whois Example
- Submitting an HTML Form from a Java Program
- Handling URL
- Using the URLConnection Objects
- Working with Datagrams
- Datagrams Server and Client
- Working with BART
- Learning about the Java.security Package
- Summary

Implementing Event Handling and Wrappers in Servlets 3.1

- Introducing Events
- Introducing Event Handling
- Working with the Types of Servlet Events
- Developing the onlineshop Web Application
- Introducing Wrappers
- Working with Wrappers
- Summary

Java Server Pages 2.3 and Expression Language 3.0

- Introducing JSP Technology
- Listing Advantages of JSP over Java Servlet
- Exploring the Architecture of a JSP Page
- Describing the Life Cycle of a JSP Page
- Working with JSP Basic Tags and Implicit Objects

- Working with Action Tags in JSP
- Exploring EL
- Using Custom Tag Library with EL Functions

Implementing Filters

- Exploring the Need of Filters
- Exploring the Working of Filters
- Exploring Filter API
- Configuring a Filter
- Creating a Web Application Using Filters
- Using Initializing Parameter in Filters
- Manipulating Responses
- Discussing Issues in Using Threads with Filters
- Summary

Java EE Design Patterns

- Describing the Java EE Application Architecture
- Introducing a Design Pattern
- Discussing the Role of Design Patterns
- Exploring Types of Patterns
- Summary

Implementing SOA using Java Web Services

- Section A: Exploring SOA and Java Web Services
- Overview of SOA
- Describing the SOA Environment
- Overview of JWS
- Role of WSDL, SOAP and Java/XML Mapping in SOA
- Section B: Understanding Web Service Specifications to Implement SOA

- Exploring the JAX-WS 2.2 Specification
- Exploring the JAXB 2.2 Specification
- Exploring the WSEE 1.3 Specification
- Exploring the WS-Metadata 2.2 Specification
- Describing the SAAJ 1.3 Specification
- Working with SAAJ and DOM APIs
- Describing the JAXR Specification
- JAXR Architecture
- Exploring the StAX 1.0 Specification
- Exploring the WebSocket 1.0 Specification
- Describing the JAX-RS 2.0 Specification
- Exploring the JSON-P 1.0 Specification
- Section C: Using the Web Service Specifications
- Using the JAX-WS 2.2 Specification
- Using the JAXB 2.2 Specification
- Using the WSEE and WS-Metadata Specifications
- Implementing the SAAJ Specification
- Implementing the JAXR Specification
- Implementing the StAX Specification