INTERSHIP TRAINING OUTLINE

1. Fundamental Programming Structures in Java

Learn the basics of the Java programming language

- Data and instructions
- Primitive and Non primitive Data Types
- Access modifiers
- Operators (Arithmetic and logical operators)
- Control Flow
    - Conditional
    - Decision making
    - Iteration
- Code Examples & Exercises Classes and Objects in Java

Input and Output in Java

- Introduction
- Standard Streams
- Keyboard Input
- File I/O Using Byte Streams
- Character Streams
- File I/O Using Character Streams
- Buffered Streams
- File I/O Using a Buffered Stream
- Keyboard Input Using a Buffered Stream
- Writing Text Files

Fundamental elements of OOP and related Java features

- Classes & Objects
- OOP Principles
    - Inheritance
    - Abstraction
    - Encapsulation
    - Polymorphism
- Instance Variables
- Class Variables
- Methods
    - Method Signatures
    - Arguments and Parameters
    - Passing Objects to Methods

    - Method Overloading
    - Method Overriding
    - Static Methods
- Constructors
- Code Examples & Exercises Object Design and Programming with Java

Managing collections of objects

- Arrays
- The Java Collections Framework
- Collections Interfaces
    - java.util.Collection
    - java.util.List
    - java.util.Map
    - java.util.Set
- Concrete Collections
    - java.util.ArrayList
    - java.util.HashMap
    - java.util.HashSet
- Iterating through Collections
    - java.util.Iterator
- Code Examples & Exercises

Threads

- Threads vs. Processes
- Creating Threads by Extending Thread
- Creating Threads by Implementing Runnable
- Advantages of Using Threads
- Daemon Threads
- Thread States
- Thread Problems
- Synchronization

Exceptions

- Introduction
- Exception Handling
- The Exception Hierarchy
- Checked Exceptions
- Advertising Exceptions with throws
- Developing Your Own Exception Classes
- The finally Block

Java Database Programming (JDBC)

- Overview
- Establishing a Connection
- Creating a JDBC Statement
- Creating a JDBC PreparedStatement
- Executing CREATE/INSERT/UPDATE Statements
- Executing SELECT Statements
- ***Practical (Hands-on Working Examples)***
- Notes on Accessing ResultSet
- ***Practical (Hands-on Working Examples)***
- Transactions
- Handling Errors with Exceptions

2. Java Platform, Enterprise Edition

Core Java EE

- Java EE 5 Platform Overview
- Java EE Platform – Distributed Multi-tiered Applications
- Java EE – Web & Business Components
- Java EE Containers – services & types
- Java EE Application Assembly & Deployment – Packaging Applications, Java EE modules
- Getting Started with Web Applications

Web application deployment

- Web application development and deployment Steps
- Configuring Web application – Web application deployment descriptor (web.xml file)
- Web Application Archive (*.WAR file) – *.WAR directory structure
- Building & Deploying Applications, Maven tool
- ***Practical (Hands-on Working Examples)***

Advanced Java EE – Web & Business Components Development

- Servlets
    - Servlet Overview
    - Life cycle of Servlet
    - Handling Client HTTP Request & Server HTTP Response
    - ***Practical (Hands-on Working Examples)***
    - Filtering Requests and Responses
    - Invoking Other Web Resources

- Uploading Files with Java Servlet Technology
- Initializing Parameters & ServletContext
  - Initializing a Servlet
  - ServletContext initialization Parameters
  - S**e**rvletContext Attributes (Context binder)
- ***Practical (Hands-on Working Examples)***
- Session Management, Request Dispatcher & Redirecting
- ***Practical (Hands-on Working Examples)***

- Java Server Pages - JSP
  - Overview of JSP
  - JSP Architecture & life cycle
  - JSP scripting elements
  - Using JavaBeans components with JSP
  - Creating custom JSP tag libraries
  - Integrating servlets and JSP with the MVC architecture.
  - ***Practical (Hands-on Examples)***
  - Implicit Objects & Standard JSP Tags
  - ***Practical (Hands-on Working Examples)***
  - Scope of JSP objects
  - ***Practical (Hands-on Working Examples)***
  - Developing Web Application with MySQL / Oracle 10g XE Database by implementing Java Beans, DAO's & MVC2 Architecture
- Resource Creation
  - Introduction & Overview
  - Java Naming and Directory Interface - JNDI
    - What is JNDI?
    - Benefits of JNDI
    - Naming Services
    - Directory Services
    - Using JNDI
    - Context Operations
    - JNDI Utility Class
    - ***Practical (Hands-on Working Examples)***
    - Naming Exceptions
  - Connection pool
    - Introduction & Overview
    - ***Practical (Hands-on Working Examples)***
- Injection
  - Resource Injection
  - Dependency Injection
- Contexts and Dependency Injection for Java EE
  - Introduction to Contexts and Dependency Injection for Java EE
    - Overview of CDI
    - About Beans

        - About CDI Managed Beans
        - Beans as Injectable Objects
        - Using Qualifiers
        - Injecting Beans
        - Using Scopes
        - Giving Beans EL Names
        - Adding Setter and Getter Methods
        - Injecting Objects by Using Producer Methods
        - Configuring a CDI Application
        - Using the @PostConstruct and @PreDestroy Annotations with CDI Managed Bean Classes
        - To Initialize a Managed Bean Using the @PostConstruct Annotation
        - To Prepare for the Destruction of a Managed Bean Using the @PreDestroy Annotation
    - Advance Contexts and Dependency Injection for Java EE
        - Packaging CDI Applications
        - Using Alternatives in CDI Applications
        - Using Specialization
        - Using Producer Methods, Producer Fields, and Disposer Methods in CDI Applications
        - Using Producer Methods
        - Using Producer Fields to Generate Resources
        - Using a Disposer Method
        - Using Predefined Beans in CDI Applications
        - Using Events in CDI Applications
        - Defining Events
        - Using Observer Methods to Handle Events
        - Firing Events
        - Using Interceptors in CDI Applications
        - Using Decorators in CDI Applications
        - Using Stereotypes in CDI Applications
- Enterprise Java Beans (EJB 3.0)
    - EJB 3.0 overview & Architecture
    - When to Use Enterprise Beans
    - Types of Enterprise Beans
    - Session Beans
        - Types of Session Beans
            - Stateful Session Beans
            - Stateless Session Beans
            - Singleton Session Beans
        - When to Use Session Beans
    - Java EE Application Assembly and Deployment – Anatomy of EJB Module & Packaging
    - ***Practical (Hands-on Working Examples)***
- Designing a Java Enterprise Application

- ***Creating an Enterprise Application Project using Eclipse + Wildfly + Oracle XE – Hands On***

- Java API for WebSocket
  - Introduction to WebSocket
  - Creating WebSocket Applications in the Java EE Platform
  - Programmatic Endpoints
  - Annotated Endpoints
  - Sending and Receiving Messages
  - Maintaining Client State
  - Using Encoders and Decoders
  - Path Parameters
  - Handling Errors
  - Specifying an Endpoint Configurator Class
  - The dukeetf2 Example Application
- Web Services
  - Introduction and overview to Web Services
  - ***Building Web Services with JAX-WS (Hands-on Working Examples)***
  - ***Building RESTful Web Services with JAX-RS (Hands-on Working Examples)***
  - ***Accessing REST Resources with the JAX-RS Client API (Hands-on Working Examples)***

3. Java Transaction, Persistence & Object Relation Mapping (ORM)

Java Transaction API

- What Is a Transaction?
- Container-Managed Transactions
- Bean-Managed Transactions
- Transaction Timeouts
- Updating Multiple Databases
- Transactions in Web Components

Java Persistence API (Hand on working examples)

- Introduction to the Java Persistence API
  - Entities creation, and persistent
  - Entity Inheritance & Entity Inheritance Mapping Strategies
  - Managing Entities
  - Querying Entities
  - Database Schema Creation
- The Java Persistence Query Language
  - Query Language Terminology
  - Creating Queries Using the Java Persistence Query Language
  - Simplified Query Language Syntax
  - Example Queries
- Using the Criteria API to Create Queries

- Overview of the Criteria and Metamodel APIs
- Using the Metamodel API to Model Entity Classes

Object Relation Mapping – Hibernate (Hands on working examples)

- Core application programing interfaces for Hibernate (Session, Transaction, Query, EntityManager, SessionFactory, Annotations)
- Mapping Persistent Classes
- Inheritance and polymorphic relationships
- Collections and Associations
- Hibernate HQL, Criteria, and JDBC/SQL queries
- Hibernate Search, Validation, Shards, and Tools technologies
- Hibernate Tools integrated development environment (IDE)
- Best practices and patterns for Hibernate developers
- Hibernate deployment, configuration, and performance tuning

4. Best Programming Practices and techniques

Introduction & overview of code smells & technical debts

Design principles – S.O.L.I.D principles

- S – Single-responsiblity principle
- O– Open-closed principle
- L – Liskov substitution principle
- I – Interface segregation principle
- D – Dependency Inversion Principle

Design Patterns

- Creational design patterns
  - Abstract Factory
  - Builder
  - Factory Method
  - Object Pool
  - Prototype
  - Singleton
- Structural design patterns
  - Adapter
  - Bridge
  - Composite
  - Decorator
  - Facade
  - Private Class Data
  - Proxy

- Behavioral design patterns
  - Command
  - Interpreter
  - Iterator
  - Mediator
  - Memento
  - Null Object
  - Observer
  - State
  - Strategy
  - Template method
  - Visitor
  - Chain of responsibility