

Nexus Corporate Training Center

FULL STACK JAVA DEVELOPER SYLLABUS

1. CORE JAVA

- 1.1. Introduction To Java
- 1.2. Java History
- 1.3. Why Java
- 1.4. Features of Java
- 1.5. Environment Setup
- 1.6. Hello World Application
- 1.7. Java Program Internal
- 1.8. JDK, JRE, JVM
- 1.9. JVM- Java Virtual Machine
- 1.10. Class, Objects, Data Types
- 1.11. Variables
- 1.12. Type Casting
- 1.13. Operators
- 1.14. Java Comments
- 1.15. Object Oriented Programming Concepts

1. Abstraction

2. Encapsulation

3. Polymorphism

4. Inheritance

- 1.16. Java Coding Standards
- 1.17. Package – Creating package, naming package
- 1.18. Java Keywords
- 1.19. Access Specifiers – Public, Private, Protected, Default
- 1.20. Static Concept – Block, Variable, Method, Class
- 1.21. Scanner Class
- 1.22. Control Statement – If, If...else, Nested if...else and Switch Case
- 1.23. Java Break Statement
- 1.24. Java Continue Statement
- 1.25. Flow Statement- for, while, do..while
- 1.26. Constructor- Default, Parameterized
- 1.27. Inheritance
- 1.28. Extends, super and this keyword
- 1.29. Method Overloading, Method Overriding
- 1.30. Interface
- 1.31. Abstract Class
- 1.32. Abstract Class vs Interface
- 1.33. Garbage Collection
- 1.34. Marker Interface
- 1.35. Serialization
- 1.36. Transient

- 1.37. Java instanceof
- 1.38. Instance initializer block
- 1.39. Volatile Keyword in Java
- 1.40. Exception Handling
- 1.41. Exception Handling- try, catch, throw, throws, finally
- 1.42. String Concept
 - String Basics
 - String Methods
- 1.43. String Buffer
- 1.44. String Builder
- 1.45. Custom Immutable Class
- 1.46. Java - Arrays
- 1.47. Collection Framework
 - Collection Basics
 - Iterator Interface
 - Methods of Collection Interface
- 1. List
 - A] Array List
 - B] Linked List
 - C] Vector
- 2. Set
 - A] Hash Set
 - B] Tree Set
- 3. Map
 - A] Hash Map
 - B] Tree Map
 - C] Hashtable
- 1.48. Collection Framework Advance Concepts
 - 1. Comparable Interface
 - 2. Comparator Interface
 - 3. Collections class
 - 4. Differentiate Comparable and Comparator
- 1.49. Java Enum
- 1.50. Date Concept
- 1.51. File Handling Concept
 - 1. File Handling Basics
 - 2. Create Folder
 - 3. Create File- .txt, .pdf, .xlsx, .docs, etc
 - 4. Write File
 - 5. Read File
 - 6. Delete File
- 1.52. Thread
 - 1. Concept, Lifecycle
 - 2. Extends Thread

- 3. Implement Runnable Interface
 - 4. Thread Priorities
 - 5. Thread Methods
 - 1.53. Multithreading
 - 1.54. Thread Synchronization
 - 1. Synchronization
 - 2. Object Locking
 - 3. Inter Thread Communication
 - 1.55. JDK 1.8.0 New Features with Hands-on
 - 1. Lambda Expression
 - 2. Functional Interface
 - 3. Default Method in Interface
 - 4. Static Method in Interface
 - 5. Method References
 - 6. Date Time API
 - 7. Stream API in details[All Advanced concepts as per Industry Practices]
 - 8. Collectors
 - 9. For each Loop
 - 10. String Joiner Class
 - 11. Parallel Sort
 - 12. Optional Class
 - 1.56. Java Mail API- Realtime Live Implementation
- Programming practices, Concepts & Interview Preparation

2. J2EE

- 2.1 JDBC
 - 1. JDBC Introduction
 - 2. JDBC Architecture
 - 3. Database Overview
 - 4. JDBC Basics
 - 5. MySQL
 - 6. Create Database
 - 7. Create Table
 - 8. Insert, Update, Delete
 - 9. Truncate
 - 10. SQL Join
 - Inner Join
 - Left Outer Join
 - Right Outer Join
 - Full Outer Join
 - 11. Aggregate Function
 - 1. MIN
 - 2. MAX
 - 3. AVG

- 4. SUM
- 5. COUNT
- 12. SQL HAVING Clause
- 13. GROUP BY
- 14. ORDER BY
- 15. SQL Aliases
- 16. SQL LIKE Operator
- 17. SQL IN
- 18. SQL NOT IN
- 19. SQL Between
- 20. SQL Null Values
- 21. SQL TOP
- 22. SQL LIMIT
- 23. SQL Stored Procedures
- 24. Java Application Using JDBC Connectivity
- 25. Handling SQL Exceptions
- 26. DriverManager
- 27. ResultSet
- 28. Connection
- 29. Statement
- 30. Prepared Statement
- 31. Collable Statement
- 32. DB Connectivity Steps
- 33. Store Image in SQL
- 34. Read Image in SQL
- 35. SQL Queries using SQLYog
- 36. JDBC CRUD Application

2.2 Servlet

- 2.2.1 Servlet Basics
- 2.2.2 Need of Server Side Programming
- 2.2.3 Servlet Life Cycle
 - 1 Init()
 - 2 Service()
 - 2.1 doGet()
 - 2.2 doPost()
 - 3 Destroy()
- 2.2.4 Servlet Hello World Application
- 2.2.5 Web.xml Structure
- 2.2.6 Servlet Directives- include(), forward(), sendRedirective()
- 2.2.7 HttpServletRequest, HttpServletResponse in Servlet
- 2.2.8 Servlet and JDBC Integration
- 2.2.9 Servlet, HTML 5, MySQL-JDBC, Apache Tomcat using Real Time Login Application

- 2.2.10 Servlet, JSP, MySQL- JDBC, Apache Tomcat using Custom CRUD Application
- 2.2.11 Servlet, JSP, MySQL- JDBC, Apache Tomcat using Custom Sign Up | Sign In Application
- 2.1 JSP
 - 2.1.1 JSP Basics
 - 2.1.2 Creating dynamic Web content with JSP
 - 2.1.3 Scriptlet
 - 2.1.4 Declaration
 - 2.1.5 Servlet, JSP, MySQL- JDBC, Apache Tomcat Login Application
 - 2.1.6 Servlet, JSP, MySQL- JDBC, Apache Tomcat using User Sign Up | Sign In Application
 - 2.1.7 Servlet, JSP, MySQL- JDBC, Apache Tomcat using Custom CRUD Application with JDK Advanced features
 - 2.1.8 Servlet, JSP, MySQL- JDBC, Apache Tomcat using Custom CRUD Application with different Date Patterns and time zones as per Industrial Practices
 - 2.1.9 Servlet, JSP, MySQL- JDBC, Apache Tomcat Using Advanced Custom Project Implementation
- Programming practices, Concept & Interview Preparation.

3. FRAMEWORK

- 3.1. Hibernate
 - 1. Hibernate Introduction
 - 2. Hibernate Basics
 - 3. Hibernate Architecture
 - 4. Hibernate Session
 - 5. Hibernate SessionFactory
 - 6. Hibernate Configuration
 - 7. Hibernate Configuration Offline
 - 8. Hibernate with HBM
 - 9. Hibernate with Annotation
 - 10. Hibernate Validator
 - 11. Hibernate CRUD
 - 12. Hibernate Association Mapping
 - 1. One to One Mapping
 - 2. One to ManyMapping
 - 3. Many to OneMapping
 - 4. Many to Many Mapping
 - 13. Hibernate Aggregation
 - 14. Hibernate Named Queries
 - 15. Hibernate Native SQL
 - 16. HQL- Hibernate Query Language
 - 17. Hibernate HCQL- Criteria Query Language

18. Hibernate Application with Mysql DB- CRUD
 19. Diff. save(), saveOrUpdate(), update(), persist() and merge()
 20. Diff. get() and load()
 21. Hibernate vs. JPA
 22. Hibernate Application with Mysql DB, Servlet, JSP-User Interface- CRUD
 23. Hibernate, MySQL, Servlet, JSP, Apache Tomcat Login Application
 24. Hibernate, MySQL, Servlet, JSP, Apache Tomcat Sign Up | Sign In Application
 25. Hibernate Application with Mysql DB, Servlet, JSP-User Interface- different Date Patterns with time zones
 26. Hibernate, MySQL, Servlet, JSP, Apache Tomcat Advanced Custom Project Implementation
 27. Programming practices, Concept & Interview Preparation.
- 3.2. Spring Framework
 1. Spring Framework Introduction
 2. Spring Basics
 3. Spring Framework Features
 4. Inversion of Control
 5. Dependency Injection
 6. Beans Scope
 - 5.1 Singleton
 - 5.2 Prototype
 - 5.3 Request
 - 5.4 Session
 - 5.5 Global-Session
 7. Auto wiring
 8. Spring Annotations
 - 3.3. Spring MVC
 1. Spring MVC Basics
 2. Spring MVC Annotations
 3. Spring MVC Hello World Application
 4. Spring MVC Restful Web Services Basics
 5. Spring MVC without Maven
 6. Apache Maven Basics
 7. Spring MVC with Maven
 8. Spring MVC Application Deployment
 9. Spring MVC Application Deployment with Loggers
 10. Spring MVC WAR File Creation and Deployment Steps
 11. Spring MVC, Hibernate, Mysql, Apache Maven, Apache Tomcat-CRUDOperation
 12. Java, Spring MVC, Spring Security, Algorithm, Hibernate, Maven, Mysql IntegrationApplication
 1. Modules-

1. Customer Login
2. Add Customer Details
3. Edit Customer Information
4. Delete Customer
5. Delete all Customer
6. View Customer List
7. Roles
8. Admin Role
9. DBA Role
10. User Role
11. Password Encryption by using Spring Security Algorithm
12. Remember Me Functionality
13. Timeout Functionality
14. Deploy on external Server
- 3.4. Spring Security
 1. Spring Security Basics
 2. Spring Security Features
 3. Spring Security Authentication
 4. Spring Security Encryption
 5. Password Encryption by using Spring Security
 6. Spring Security Login Form
 7. Spring Security Authentication
 8. Spring Security Maven Dependency
 9. Spring Security CRUD- Signup and Login Application
 10. Spring Security OAuth2 RealTime Implementation
 11. Spring Boot End Points Security
 12. Spring Boot, Spring Security- Restful Web Services with Advanced Authentication
 13. Single Sign-On with Spring Boot & Okta | OAuth 2.0
 14. Spring Boot with JWT Token base authentication
 15. Spring Boot with Spring Security using BCrypt Algorithm Role base authorization
- 3.5. Spring Boot
 1. Spring Boot Introduction
 2. Spring Boot Basics
 3. Why Spring Boot
 4. Spring Boot Features
 5. Main Goal of Spring Boot
 6. Creating Project- Spring Initializer
 7. Spring Boot Hello World Application
 8. Spring Boot Annotations
 9. Spring Boot Devtool Dependency- Without Restart application will see changes on browser
 10. Spring Boot Actuator

11. Spring Boot Build Systems
12. Spring Boot Code Structure
13. Spring Boot Tomcat Custom Port Number
14. Spring Boot with RESTFul Service
15. Spring Boot with Logger
16. Deploy Spring Boot Applications on External Tomcat Server
17. Spring Boot – Common Application Properties (application.properties)
18. Spring Boot, Hibernate with RESTFul Web Service
19. Spring Boot, Spring Data with RESTFul Web Service
20. Spring Boot, Hibernate CRUD Application
21. Spring Boot, Spring Data JPA CRUD Application
22. Spring Boot, Spring- JDBCTemplate CRUD
23. Spring Boot, Hibernate, MySQL, Project Lombok CRUD Application
24. Spring Boot with XML Payload using Postman API
25. Spring Boot with JSON Payload using Postman API
26. Spring Boot Global Custom Validations
27. Spring Boot Global Custom Exception Handling
28. Spring Boot Code Debug Practices
29. Spring Boot Application connectivity with Multiple Databases
30. Spring Boot CORS Support
31. Spring Boot with Hibernate, Project Lombok, Swagger UI
32. Spring Boot with Hibernate HRM Application, Swagger UI
33. Spring Boot Local Network Chat Application
34. Spring Boot QR Code Generator Hands on Session
35. Spring Boot QR Code Reader Hands on Session
36. Spring Boot with Spring Security- JWT
37. Spring Boot Advanced Research and Development with latest release as per Industry Practices