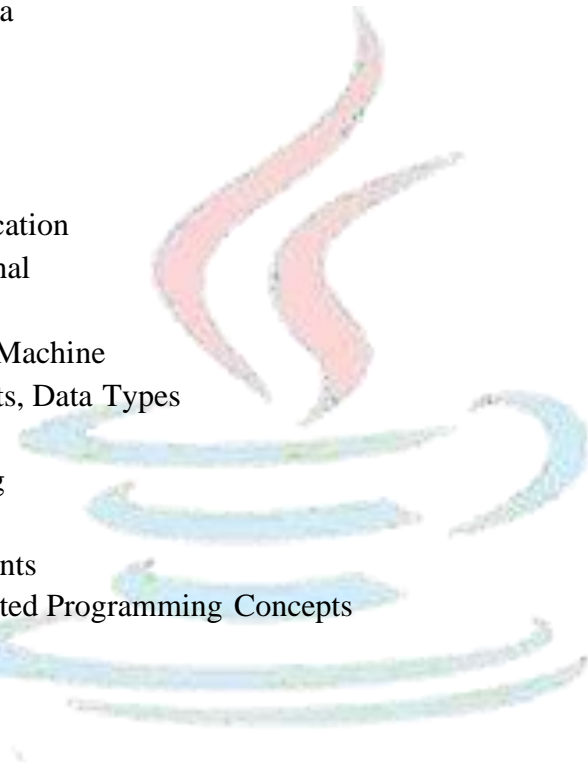


## 1. CORE JAVA

- 
- 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
  - 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. Callable 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 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 ManyMapping
  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, MySQL, Servlet, JSP, Apache Tomcat Advanced Custom Project Implementation
- Programming practices, Concept & Interview Preparation.

## 3.2. Spring Framework

1. Spring Framework Introduction
2. Spring Basics
3. Inversion of Control
4. Dependency Injection
5. Beans Scope
  - 5.1 Singleton
  - 5.2 Prototype
  - 5.3 Request
  - 5.4 Session
  - 5.5 Global-Session
6. Auto wiring
7. 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- CRUD Operation
12. Java, Spring MVC, Spring Security, Algorithm, Hibernate, Maven, Mysql Integration Application

#### 1. Modules-

1. Customer Login
2. Add Customer
3. Edit Customer Information
4. Delete Customer
5. View Customer List
6. Roles
7. Admin Role
8. DBA Role
9. User Role
10. Password Encryption by using Spring Security Algorithm
11. Remember Me Functionality

### 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

### 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 Runners
14. Spring Boot Tomcat Custom Port Number
15. Spring Boot with RESTFul Service
16. Spring Boot with Logger
17. Deploy Spring Boot Applications on External Tomcat Server
18. Spring Boot – Common Application Properties (application.properties)
19. Spring Boot, Hibernate with RESTFul Web Service
  20. Spring Boot, Spring Data with RESTFul Web Service
  21. Spring Boot, Hibernate CRUD Application
  22. Spring Boot, Spring Data JPA CRUD Application
  23. Spring Boot, Spring- JDBCTemplate CRUD
  24. Spring Boot, Hibernate, MySQL, Project Lombok CRUD Application
  25. Spring Boot with XML Payload using PostmanAPI
  26. Spring Boot with JSON Payload using PostmanAPI
  27. Spring Boot Global Custom Validations
  28. Spring Boot Global Custom Exception Handling
  29. Spring Boot Code Debug Practices
  30. Spring Boot Application connectivity with Multiple Databases
  31. Spring Boot CORS Support
  32. Java QR Code Generator Basics
  33. Spring Boot QR Code Generator Hands on Session
  34. Spring Boot QR Code Reader Hands on Session
  35. Spring Boot Advanced Research and Development with latest release

### **3.6. Spring Data- JPA**

1. Spring Data JPA Basics
2. Spring Data JPA Introduction
3. Spring Data JPA Architecture
4. Spring Data JPA application.properties file
5. JPA Repository
6. CRUD Repository
7. Spring Boot, Spring Data JPA CRUD using PostmanAPI
8. Spring Boot, Spring Data JPA, Project Lombok CRUD using Postman API Hands onSession

### **3.7. JUnit- Mockito Framework**

1. What is Unit testing
2. What is JUnit
3. JUnit Rules
4. JUnit Arithmetic Application Example
5. Mockito Introduction
6. Mockito Annotations
7. Mockito Setup
8. What is Mocking
9. Need for mocking
10. Methods of Mockito
11. Mockito mock() method
12. Mockito when() method
13. Mockito verify() method
14. Mockito doNothing() method

15. Mockito times() method
16. PowerMock
17. JUnit Mockito Hibernate CRUD
18. JUnit Mockito JPA CRUD
19. JUnit Mockito JdbcTemplate CRUD

### **3.8. Spring- JdbcTemplate**

1. Spring JdbcTemplate Introduction
2. RowMapper
3. Builder
4. Spring Boot, JdbcTemplate using Postman API Hands on Session
5. Industrial Practices for Spring Boot JdbcTemplate
6. Spring Boot JdbcTemplate CRUD Application
7. Spring Boot JdbcTemplate SignUp | SignIn Application
8. Real Time Signup and Signin Application using JdbcTemplate
9. Advanced Research and Development for JdbcTemplate using Spring Boot

## **4. WEB SERVICES**

- 4.1. SOAP Web Service
- 4.2. RESTful Web Service

## **5. Design Pattern**

- 5.1. Singleton Design Pattern
- 5.2. Builder Design Pattern
- 5.3. Factory Pattern
- 5.4. Abstract Factory Pattern
- 5.5. Prototype Pattern
- 5.6. Facade Pattern
- 5.7. Adapter Pattern
- 5.8. Filter Pattern
- 5.9. Composite Pattern
- 5.10. Decorator Pattern
- 5.11. Proxy Pattern
- 5.12. Iterator Pattern
- 5.13. Bridge Pattern
- 5.14. Flyweight Pattern
- 5.15. MVC Pattern

## **6. MICROSERVICES**

- 6.1. Microservices Introduction
- 6.2. Microservices Architecture
- 6.3. Microservices Project Structure
- 6.4. Microservices Frameworks
- 6.5. Spring Boot with Microservices Basics
- 6.6. Spring Boot with Microservices Coding Standards



- 6.7. Spring Boot with Microservices Builder Design Pattern
- 6.8. Spring Boot with Microservices QR Code Generator
- 6.9. Application.properties vs application.yml file
- 6.10. Spring Boot, Hibernate, Maven, MySQL CRUD Application- POSTMAN API
- 6.11. Spring Boot, Spring Data- JPA, Maven, MySQL CRUD Application- POSTMAN API
- 6.12. Spring Boot, Spring Data- JdbcTemplate, Maven, MySQL CRUD Application
- 6.13. Spring Boot, ORM, Microservices with Project Lombok
- 6.14. Spring Boot, CORS, Spring Data[JdbcTemplate], Microservices CRUD with Project Lombok using React JS
- 6.15. Creating Multiple Services using Spring Boot
- 6.16. Logger with Microservices
- 6.17. Rest Template and RestTemplateVO Implementation
- 6.18. Set port number and application name for each service
- 6.19. Service Registry- Eureka Server
- 6.20. API Gateway for traverse centralize request
- 6.21. Hystrix DashBoard
- 6.22. Zipkin
- 6.23. Sleuth
- 6.24. Trace Application- System Health
- 6.25. Trace Fallback controller
- 6.26. Trace Services are up/down using controller service
- 6.27. Trace Dependent Services
- 6.28. Trace Controller | Methods and Services how many times used/hit
- 6.29. Advanced Level Research and Development for Microservices
- 6.30. Microservices HRM/Payroll Application
- 6.31. Microservices Customer | Product | Payment Order Application

## 7. WEB DESIGN

- 7.1. HTML 5
- 7.2. CSS 3
- 7.3. JavaScript
- 7.4. Bootstrap
- 7.5. Payment Gateway Integration
- 7.6. Domain and Hosting
- 7.7. Website Deployment and Go Live

## 8. React JS

- 8.1. React JS Basics
- 8.2. React JS Features
- 8.3. React JS Setup and Hello World Application
- 8.4. React JS JSX
- 8.5. React JS Component
- 8.6. React JS State
- 8.7. React JS Props
- 8.8. React JS Constructor
- 8.9. React JS Lifecycle
- 8.10. React JS Events

- 8.11. React JS Router
- 8.12. React JS Forms
- 8.13. React JS Tables
- 8.14. React JS Portals
- 8.15. React JS ES6
- 8.16. React JS CSS
- 8.17. React JS Hook
- 8.18. React JS and Back End Integration
- 8.19. React JS Using Back End CRUD Application
- 8.20. React JS, JDK 1.8, Spring Boot, Hibernate, Maven, MySQL, Microservices, Project Lombok- Project Implementation

## 9. AWS

- 9.1 What is AWS
- 9.2 Introduction to AWS Services
- 9.3 Amazon Cloud Services
- 9.4 Create EC2 Instance in AWS
- 9.4 Deploy Spring Boot Application on AWS | Elastic Beanstalk
- 9.5 AWS Advanced Research and Development- Live Deployment

## 10. INDUSTRIAL ADVANCE CONCEPTS

- 6.1. SVN
- 6.2. Jenkin
- 6.3. JIRA
- 6.4. Postman API
- 6.5. GIT
- 6.6. Maven
- 6.7. JUnit
- 6.8. JSON
- 6.9. XML
- 6.10. Spring Boot JMS ActiveMQ | RabbitMQ
- 6.11. Apache Kafka
- 6.12. JUnit- Mockito
- 6.13. REST API Documentation Tool | Swagger UI
- 6.14. Code Analysis using SonarQube
- 6.15. Code Coverage With Jacoco
- 6.16. Spring Boot with Docker
- 6.17. Spring Boot Kubernetes
- 6.18. Principles of SOLID
- 6.19. DevOps- CI/CD
- 6.20. Agile Methodology
- 6.21. Data Structure and Algorithms

## **11. INDUSTRIAL SOFTWARES**

1. JDK 1.8
2. Eclipse
3. IntelliJ IDEA
4. Apache Tomcat
5. Apache Maven
6. MySQL
7. SQLYog
8. Node.js
9. Sublime
10. Visual Studio Code

## **12. INTERVIEW QUESTIONS & PREPARATIONS**

- Developer Life
- Programming practices
- Concepts
- Interview Preparation
- Career Guidance
- Professional Consultation