

Java Full Stack Development Syllabus

Module 1: Core Java

1. Features of Java Language

- ❖ Simple.
- ❖ Object-Oriented.
- ❖ Platform Independent.
- ❖ Portable.
- ❖ Robust.
- ❖ Secure.
- ❖ Interpreted.
- ❖ Multi-Threaded

2. Language Fundamentals

- ❖ Identifiers, literals
- ❖ Data types -
- ❖ Primitives and bucket concept
- ❖ Arrays
- ❖ Java coding standards
- ❖ Downloads and Installations -

JDK - 1.8

Eclipse – Photon or Oxygen

3. Operators and Assignments

- ❖ Increment & Decrement operators.
- ❖ Arithmetic operators.
- ❖ String concatenation operators.
- ❖ Relational operators
- ❖ Equality operators
- ❖ Bitwise operators (&, |)
- ❖ Short circuit operators (&&, ||)
- ❖ instance of operators
- ❖ Assignment operator
- ❖ Conditional operator
- ❖ new operator
- ❖ [] operator

4. Java inside

- ❖ JDK, JRE, JVM
- ❖ Memory Organisation in Java
 - 5 memory areas viz Method, Stack, Heap, PC, Native
- ❖ Garbage Collection in Java
- ❖ Loading, Linking, Initialisation
- ❖ Class loader
- ❖ Class Loader types and properties

5. Declarations and Access Control

- ❖ Java project, package, class
- ❖ Members of class
- ❖ static vs instance
- ❖ constructor details
- ❖ Modifiers - 12 (4+8)

6. Flow control

- Selective
 - if, if-else, if-else ladder, nested if-else, ternary operator, switch
- Iterative
 - while
 - do-while
 - for
 - for-each
- Transfer
 - break
 - continue
- Arrays examples

7. Exception Handling

- ❖ Exception and Error
- ❖ Hierarchy of Throwable
- ❖ checked vs unchecked, partially checked, errors
- ❖ How to handle exceptions

- Try-catch-finally blocks
- Throw
- Throws
- ❖ Code flow in try catch blocks
- ❖ Top 10 exceptions with examples
- ❖ Custom Exceptions
- ❖ Exception improvements in 1.7

8. OOPS - Object Oriented Programming Concepts

- Inheritance
 - Definition, types, Diamond problem, solution
- Abstraction
 - Abstract class and Interface details
- Encapsulation
 - Meaning, 12 modifiers, their applicability
- Polymorphism
- Overloading
 - Rules for signature
- Overriding
 - Rules for signature, Return Type, Modifiers, Exceptions
- Code scenarios for OOPS

9. JFC – Java Fundamental Classes

- ❖ String Class
- ❖ StringBuffer, StringBuilder
- ❖ Object Class
- ❖ Wrapper Classes
- ❖ Thread
- ❖ File related classes

10. Java 5 features

- ❖ Static imports
- ❖ for-each loop
- ❖ Var-arg methods
- ❖ ENUM
- ❖ AutoBoxing-Unboxing

11. File Handling and IO Package

- ❖ File
- ❖ BufferedReader
- ❖ PrintWriter
- ❖ Serialisaion
- ❖ serialVersionUID
- ❖ Custom Serialisaion
- ❖ Externalisation

12. Inner Classes

- ❖ static inner class
- ❖ member inner class
- ❖ local inner class
- ❖ nested inner class
- ❖ calling variables and methods of inner classes

13. Multithreading

- ❖ Thread class, Runnable Interface
- ❖ Thread scheduler
- ❖ methods in Thread Class - start(), run(), join(), yield(), sleep(), and others
- ❖ Thread life Cycle
- ❖ Synchronisation
- ❖ Class level lock and Object level lock
- ❖ Inter-Thread communication - wait(), notify(), notifyAll()

14. Collection Framework

- Limitations of legacy classes and Arrays
- List and its classes
- Set and its classes
- Usage, utilities, backend DS, methods of each collection classes
- Collections class and its utility methods
- Cursors in java
 - Enumeration
 - Iterator
 - ListIterator
- Fail fast and Fail Safe (ConcurrentModificationException)
- Map Interface
- HashSet vs HashMap
- HashSet add method and HashMap put method
- Internal working of put() and get() methods
- Map, Entry interface methods

15. Java 8 features and Usage

- ❖ Lambda expression
- ❖ Stream API
- ❖ forEach method
- ❖ Functional Interface
- ❖ default and static methods in Interfaces.
- ❖ Other improvements
- ❖ Methods of Stream Class
- ❖ Collection API improvements.
- ❖ Java Time API
- ❖ Concurrency API improvements
- ❖ Java IO improvements
- ❖ Various Examples of stream

16. Design Pattern

- ❖ Creational
 - Factory Method Pattern
 - Abstract Factory Pattern
 - Singleton Pattern
 - Prototype Pattern
 - Builder Pattern
 - Object Pool Pattern
- ❖ Structural
 - Adapter Pattern
 - Bridge Pattern
 - Composite Pattern
 - Decorator Pattern
 - Facade Pattern

- Flyweight Pattern
- proxy Pattern
- ❖ Behavioural
 - Chain of Responsibility Pattern
 - Command Pattern
 - Interpreter Pattern
 - Iterator Pattern
 - Mediator Pattern
 - Memento Pattern
 - Observer Pattern
 - State Pattern
 - Strategy Pattern
 - Template Pattern
 - Visitor Pattern
 - Null Object

Module 2: DBMS – Database Management System

1. SQL

- ❖ DDL
- ❖ DML
- ❖ MySql database
- ❖ MySql database default functions - max, min, avg, distinct, limit, offset, count, rank
- ❖ SQL joins
 - inner join
 - left join

- right join
- outer join
- ❖ Frequently asked queries in interviews
- ❖ SQL injection
- ❖ Prepared statements and Stored Procedures
- ❖ Complex queries

2. JDBC – Java Database Connectivity

- ❖ Connection steps
- ❖ MySQL connector jar importance
- ❖ JDBC classes and interfaces
- ❖ JDBC fetch and insert query
- ❖ Driver types

3. Hibernate

- Limitations of JDBC
- hibernate.cfg.xml file
- mapping types - xml, annotations
- Hibernate Inheritance
 - Table per hierarchy, Table per class, Joined
- Hibernate Mappings-
- Collection mapping (List, Set, Map)
- One to One Mappings
- One to Many Mappings
- Many to Many Mappings
- Hibernate Cache

- Primary
- Secondary
- Query Cache
- Session interface methods save(), persist(),get(), load(), save() saveOrUpdate(), merge(), evict(), clear(), close()
 - JPA methods
 - Hibernate methods
- Cascading strategies --> All, Detach, Remove, Merge -----
-> Delete, delete_orphan,
- HQL language
- Exceptions in Hibernate
- Query, Criteria and Criterion Interface
- Hibernate Pagination
- Named Query
- Hibernate Projections
- Lazy Loading

Module 3: JavaScript

JavaScript Course Syllabus

- Introduction
- What is JavaScript?
- JavaScript Features
- JavaScript Libraries
- Understanding HTML, CSS, JavaScript
- JavaScript Version
- ES5
- ES6/ES2015

- ES7/ES2016
- ES8/ES2017
- Data Types
- Number
- String
- Boolean
- Undefined
- Null
- Variable Mutation
- Basic Operator
- Operator Precedence
- Shorthand operator
- Problem statement
- Decision making statement-if else
- if else - example
- Comparing the problem statement solution

- Switch statement - example
- Boolean logic

- Boolean logic example
- Ternary operator
- Ternary operator example

- Tips

- Functions
- Function statement
- Function Statement Example
- What is Array
- Real world Problem Ecommerce

- Real world problem transport

- Space and the Complexity
- Built in function Complexity

- examples of Array
- Objects, When to use it
- Objects and Methods
- Loops and Iteration
- For Loop
- Continue and Break Statement

ES6 JavaScript Course Syllabus

- History of JavaScript
- Features
- let & const and its example
- Arrow Functions
- Alternative
- Tips and Arrow Functions
- Exports and Imports
- Tips for exports and imports
- Classes
- Classes example
- Inheritance
- Spread and rest Parameter
- Destructuring

Module 4: React

SECTION 1: WHAT IS REACT JS?

- React JS Introduction
- Advantages of React JS
- Work flow of React JS

- Scope of React JS

SECTION 2: OVERVIEW OF JSX

- Introduction of Virtual DOM.
- Difference between JS and JSX.
- React Components overview
- Containers and components
- What is Child Components?
- What is Namespaced components?
- What are the JavaScript expressions available in JSX?

SECTION 3: REACT JS ENVIRONMENT SETUPS

- Node setup
- How to use NPM?
- How to create package.json and purpose of it?
- ES6 Introduction and features.
- Webpack Overview
- Best IDE for React JS and How to write optimized code in React JS? ➤ React JS browser plugins overview.

Real-time Practicals

- NPM Installation by locally and Globally
- Create a Basic App with React JS and other Supported NPM

SECTION 4: A REAL-TIME APPLICATION BY USING REACT JS

- Create a React component with JSX template. ➤ How to create Nested Components?
- What is React JS render?
- React Props overview.
- Introduction of Props validation with data types.
- Flow of States, Initialize states and update states.

Real-time Practicals

- Create a Small React Module
- Use All the states in in the created Application.

SECTION 5: REACT JS FORMS AND UI

- Lists of Form components.
- Setup Controlled and Uncontrolled form components.
- Control Input elements.
- How to set default values on all formats of Input elements.

- React JS Form validations.
- How to write Styles?
- Animations overview

Real-time Practicals

- Create a React Form.
- Client-side form validation.
- Applying form components.
- Submit and Rest the form.
- **SECTION 6: REACT JS COMPONENT LIFE CYCLES OVERVIEW**
 - Initial Render
 - Props Change
 - Stage Change
 - Component willMount

 - Component didMount

 - Component Unmount

Real-time Practicals

- Applying Different Lifecycles in the Application.
- When to choose Appropriate lifecycles.

SECTION 7: ROUTING IN REACT JS AND OTHER JS CONCEPTS

- Single Page Application Overview.
- How to configure React Router?
- History of Router
- How to Handle Conditional statement in JSX?
- IIFE in JSX for complex logic overview.

Real-time Practicals

- Create a Single Page Application.
- Applying Routing.
- Dynamically render the components based on the url.

SECTION 8: EVENT HANDLING IN JSX

- onBlur, onKeyUp, onChange and other useful primary events in React JS.
- How to Sharing events between the components?

Real-time Practicals

- Communicate Data between components.
- Applying all lists of events.

SECTION 12: UNIT TESTING OVERVIEW

- What are the necessary Tools required for Unit Testing?
- React Unit Testing overview
- Introduction to JEST.
- How to Test React Component?
- How to Test React Router?

SECTION 14: REACT SERVER INTEGRATION & DEPLOYMENT

- https
- httpster
- npm

Module 4: Advance Java and Web Technologies

1. Initial Web Tech

- ❖ Servlet
- ❖ web.xml details
- ❖ Servlet life cycle
- ❖ HTML
- ❖ JSP
- ❖ JSP life cycle

2. Spring

- ❖ Spring framework introduction
- ❖ DI and IoC
- ❖ IoC containers
- ❖ Spring block diagram
- ❖ Spring Hello World examples
- ❖ Ways to configure Spring
- ❖ Bean properties

- id
- scope
- custom methods –
 - init,
 - destroy
 - default-init
 - default-init
- autowire
 - by type
 - by name
- lazy-init
- properties
- ❖ Dependency Injection
 - Constructor injection
 - Setter injection
- ❖ Exceptions in spring
- ❖ Spring MVC
 - Model
 - View --JSPs
 - Controller
- ❖ Spring Controller imp annotations
 - @Component ,@Controller, @Service, @Repository
 - @RestController
- ❖ Spring AOP
- ❖ Exception Handling in Spring MVC
- ❖ Spring Batch Processing

3. Spring Boot

- ❖ Need of Boot
- ❖ starter on Boot
- ❖ Autoconfigurations
- ❖ Dev tool
- ❖ DB connection with Boot
- ❖ Profiling in Spring Boot
- ❖ Unit testing with Mokito
- ❖ Actuaters

4. Microservices -

- ❖ Monolythic app
 - ❖ Need of Microservices
 - ❖ writing a Microservice
 - ❖ CRUD operation
 - ❖ Microservice with Spring Boot
-

Module 5: Developers Tools and imp Technologies

- 👉 Maven
- 👉 Git
- 👉 Jira
- 👉 Tomcat
- 👉 Junit/TestNG
- 👉 Mockito
- 👉 Jenkins
- 👉 SONAR Cube introduction
- 👉 Agile
- 👉 Linux and Shell Script
- 👉 Deployment knowledge (release, sprint, Envs)
- 👉 Docker and Kubernetes
