Target Ready 2024 batch

Duration:

- 90 hours
- 30 sessions
- 3 hours per session
- 3 sessions per week
  - Mondays and Thursdays between 5:30 PM and 8:30 PM
  - Saturdays between 10:00 AM and 1:00 PM

Distribution of time

- Java (intermediate to slightly advanced) - 14 sessions
- Kotlin - 4 sessions
- DSA - 6 sessions
- Introduction to React js - 4 sessions
- Introduction to messaging and Kafka - 2 sessions

Detailed session-wise outline

Session 1: Java Programming Fundamentals

- Review of basic Java syntax and concepts
- Data types, variables, and operators
- Control flow: if-else, loops (for, while, do-while)
- Methods and functions
- Introduction to debugging techniques

Session 2: Object-Oriented Programming (OOP) in Java

- Principles of OOP: encapsulation, inheritance, polymorphism, and abstraction
- Classes and objects
- Constructors and method overloading
- Inheritance and method overriding
- Introduction to access modifiers

Session 3: Advanced OOP Concepts

- Interfaces and abstract classes
- Packages and access specifiers

- Composition vs. inheritance
- Enumerations and nested classes
- Best practices in OOP design

Session 4: Exceptions Handling

- Understanding exceptions and error handling
- Handling exceptions using try-catch blocks
- Throwing and propagating exceptions
- Multiple catch blocks and finally block
- Best practices for exception handling

Session 5: Input-Output (IO) Operations

- Overview of Java IO API
- Working with streams: InputStream, OutputStream, Reader, Writer
- File input and output operations
- Reading and writing text files
- Binary file handling in Java

Session 6: File Handling in Java

- Working with directories and files
- File manipulation: creating, deleting, renaming files
- File searching and filtering
- Managing file permissions and attributes
- Handling file I/O errors and exceptions

Session 7: Introduction to Collections Framework

- Overview of Java Collections Framework (JCF)
- Lists: ArrayList, LinkedList
- Sets: HashSet, TreeSet
- Maps: HashMap, TreeMap
- Iterators and enhanced for loops

Session 8: Streams and Functional Interfaces

- Introduction to streams in Java
- Stream operations: filter, map, reduce
- Working with functional interfaces
- Lambda expressions in Java

- Stream API best practices

Session 9: More on Collections

- Collection interfaces: Collection, List, Set, Map
- Collection algorithms and utility classes
- Sorting and searching collections
- Customizing and optimizing collections
- Practical exercises and case studies

Session 10: JDBC Part 1

- Overview of JDBC and database fundamentals
- Connecting to databases using JDBC
- Executing SQL queries and updates
- Handling result sets and metadata
- Best practices in JDBC programming

Session 11: JDBC Part 2

- Prepared statements and stored procedures
- Batch processing and transaction management
- Connection pooling and data source configuration
- JDBC drivers and database-specific features
- Error handling and debugging in JDBC applications

Session 12: Introduction to JPA (Java Persistence API)

- Understanding ORM (Object-Relational Mapping)
- Introduction to JPA and its architecture
- Setting up JPA with Hibernate or EclipseLink
- Mapping entities to database tables
- Performing basic CRUD operations using JPA

Session 13: Advanced JPA Concepts

- Entity relationships: One-to-One, One-to-Many, Many-to-One, Many-to-Many
- Lazy loading vs. eager loading
- Fetch strategies and performance optimization
- Transaction management in JPA
- Using JPQL (Java Persistence Query Language)

Session 14: JPA Querying and Optimization

- Advanced querying with JPQL and Criteria API
- Named queries and dynamic queries
- Caching and query optimization techniques
- Handling concurrency issues in JPA applications
- Profiling and performance tuning with JPA

Session 15: Introduction to Kotlin

- Overview of Kotlin
- History and purpose of Kotlin
- Setting up Kotlin development environment (IDE installation)
- Basic syntax and structure of Kotlin
- Variables and data types
- Basic operations and expressions

Session 16: Control Flow and Functions

- Conditional statements (if, else, when)
- Looping constructs (for, while, do-while)
- Introduction to functions in Kotlin
- Function declaration and invocation
- Parameters and return types
- Scope and visibility of variables

Session 17: Object-Oriented Programming in Kotlin

- Introduction to object-oriented programming concepts
- Classes and objects in Kotlin
- Properties and methods
- Constructors and initialization
- Inheritance and polymorphism
- Interfaces and abstract classes

Session 18: Advanced Kotlin Concepts

- Exception handling in Kotlin
- Null safety and handling nullable types
- Collections and arrays
- Lambdas and higher-order functions
- Extension functions and properties

- Coroutines (basic overview)
- Kotlin Standard Library overview

Session 19: Introduction to Data Structures and Algorithms

- Overview of the course objectives and structure
- Importance and relevance of Data Structures and Algorithms
- Basic terminologies: Data, Data Structures, Algorithms
- Characteristics of good algorithms
- Big O notation and time complexity analysis
- Introduction to different types of data structures: Arrays, Linked Lists, Stacks, Queues

Session 20: Arrays and Linked Lists

- Review of arrays: Declaration, Initialization, Accessing elements, Insertion, Deletion, Searching
- Advantages and limitations of arrays
- Introduction to Linked Lists: Singly Linked Lists, Doubly Linked Lists
- Operations on Linked Lists: Insertion, Deletion, Searching
- Comparison between arrays and linked lists

Session 21: Stacks and Queues

- Introduction to Stacks: Definition, Operations (Push, Pop, Peek), Applications
- Implementation of Stack using arrays and linked lists
- Introduction to Queues: Definition, Operations (Enqueue, Dequeue), Applications
- Implementation of Queue using arrays and linked lists
- Comparison between stacks and queues

Session 22: Trees and Binary Trees

- Introduction to Trees: Terminologies (Root, Node, Leaf, Parent, Child), Types of Trees
- Introduction to Binary Trees: Definition, Properties, Operations (Traversal, Insertion, Deletion)
- Binary Tree Traversal: Inorder, Preorder, Postorder
- Special types of Binary Trees: Binary Search Trees (BST), Balanced Binary Trees (AVLTrees)

Session 23: Sorting Algorithms

- Introduction to Sorting: Importance, Terminologies (Stable Sorting, Comparison Sorting)
- Bubble Sort: Algorithm, Implementation, Time Complexity
- Selection Sort: Algorithm, Implementation, Time Complexity

- Insertion Sort: Algorithm, Implementation, Time Complexity
- Comparison of sorting algorithms

Session 24: Searching Algorithms and Conclusion

- Introduction to Searching: Linear Search, Binary Search
- Linear Search: Algorithm, Implementation, Time Complexity
- Binary Search: Algorithm, Implementation, Time Complexity
- Comparison of searching algorithms
- Review of course content and key takeaways
- Discussion on further learning resources and career opportunities in DSA

Session 25: Understanding React.js Fundamentals

- Introduction to React.js and its significance in modern web development
- Overview of JavaScript ES6 features used in React.js
- Setting up the development environment (Node.js, npm, create-react-app)
- Introduction to JSX (JavaScript XML)
- Understanding components in React.js (functional vs. class-based components)
- State and props in React.js components

Session 26: React.js Component Lifecycle and State Management

- Understanding the lifecycle of React.js components
- Exploring component lifecycle methods (e.g., componentDidMount, componentDidUpdate)
- Handling events in React.js components
- Introduction to state management in React.js using useState and useEffect hooks
- Managing state across components using props and lifting state up

Session 27: React.js Routing and Hooks

- Introduction to React Router for handling client-side routing
- Configuring routes in a React.js application
- Implementing navigation in React.js using Link and NavLink components
- Introduction to React.js hooks (useState, useEffect, useContext)
- Implementing custom hooks for reusable logic

Session 28: Advanced Topics in React.js

- Introduction to advanced React.js concepts (e.g., context API, higher-order components)
- Working with forms in React.js (controlled vs. uncontrolled components)

- Introduction to styling in React.js (inline styles, CSS modules, styled-components)
- Best practices and tips for building scalable React.js applications
- Q&A and Recap of the course content

Session 29: Understanding Messaging Systems and Introduction to Kafka

- Introduction to Messaging Systems
  - Definition and significance
  - Types of messaging systems: synchronous vs. asynchronous
  - Use cases and applications in real-world scenarios
- Fundamentals of Apache Kafka
  - Overview of Kafka: features and benefits
  - Kafka architecture: topics, partitions, brokers, producers, consumers
  - Basic concepts and terminologies
- Setting Up Kafka Environment
  - Installation of Kafka on local machines
  - Configuration and setup for development environment
  - Practical exercises to familiarize with Kafka basics

Session 30: Core Concepts and Hands-On Practice with Kafka

- Kafka Producer and Consumer
  - Understanding producers and consumers in Kafka
  - Writing simple Kafka producers and consumers
  - Message publishing and consumption: demonstration and exercises
- Kafka Topics and Partitions
  - Deep dive into Kafka topics and partitions
  - Partitioning strategies and their implications
  - Hands-on activities to work with topics and partitions
- Kafka Ecosystem Overview
  - Introduction to Kafka ecosystem components (Kafka Connect, Kafka Streams)
  - Overview of their functionalities and use cases
- Real-World Examples and Use Cases
  - Case studies showcasing how Kafka is used in various industries
  - Discussion on handling real-time data streams and event-driven architectures