Full Stack with Java Develoment

JavaCoders

www.codewithashu.in

Trainer : Ashutosh Pandey

Core Java Syllabus: Core Java Made Easy!! Learn more than 1000+ MCQ interview questions with latest versions of JDK. At the end of session you will master in core java.

1. Java Development Kit
2. Java Platform Independency
3. Object Oriented Programming Introduction
4. The 4 Object Oriented Principles
5. Encapsulation
6. Inheritance
7. Abstraction
8. Polymorphism
9. Building Blocks of a Java Program
10. Methods
11. Introduction to Java and OOPS
12. Installing Java
13. Installing Eclipse
14. Configuring JDK in eclipse
15. Downloading the completed projects
16. Create a Hello World Program
17. Hello World Explained
18. First Java Program
19. Static and Non Static Contexts
20. Static Blocks
21. Static Methods
22. Static Variables
23. Static Members
24. Non Static Members
25. Create Non Static Variables Blocks and Constructor
26. Static versus Non Static Blocks
27. Default Constructor

28. Create a object reference

29. Create object reference in a static block

30. Create a static reference directly

31. The this Keyword

32. Create a reference in a static method

33. Invoking a non static method

34. Static versus Non Static a Summary

35. Non Static Members

36. Data Type Introduction

37. Data Type In Action

38. Type Casting

39. Explicit Down casting

40. Up casting In Action

41. Type Casting In Action Beyond Range

42. Data Types

43. Variables

44. Wrapper Classes Introduction

45. Primitives and Objects

46. Primitives and String

47. String and Objects

48. Wrapper Type Constructors

49. Wrap up the wrapper types

50. Command Line Arguments

51. Wrapper Classes

52. Increment and Decrement Operators

53. Arithmetic Operators

54. String Concatenation Operator

55. Relational Operators

56. bitwise operators

57. short circuit operators

58. assignment operator

120. Methods Signatures and multiple interfaces

121. Interfaces vs Abstract Classes

122. Compile Time Polymorphism Introduction

123. Compile Time Polymorphism In Action

124. Runtime Polymorphism

125. Runtime Polymorphism In Action

126. Interfaces

127. Using Interfaces

128. Object Casting

129. Polymorphism

130. More Programs - Overloading and Auto Promotion

131. Auto Promotions and Object Types

132. Overriding and Static Method

133. Variables and Overriding

134. Encapsulated Class

135. Advantages

136. Encapsulation

137. Encapsulation

138. Encapsulate the Event Management Application

139. Abstraction in Event Management Use case

140. Runtime Polymorphism in Action

141. Exception Handling Introduction

142. Exception while dividing numbers

143. Exception while parsing a String

144. ArrayIndexOutOfBoundException

145. NullPointerException

146. Exception Class Hierarchy and Handling

147. Handling Exceptions

148. Multiple Catch Blocks

149. Exceptions and Inheritance

150. Handling Checked Exceptions

151. Using a finally block
152. Using a throws keyword
153. Using a throw keyword
154. Creating Custom Exceptions
155. Exception Handling
156. Assertions
157. Assertions Hands On
158. Exception Handling
159. Single Threaded Example
160. Multi-Threading in Action
161. Sleep Method
162. Join Method
163. Calculating Time
164. Thread Identity
165. Thread Priority
166. Implementing Runnable Interface
167. Yield Method Demo
168. Interrupt Method Demo
169. Synchronization
170. Synchronization Demo
171. Class Level Lock
172. Class Level Lock Demo
173. Synchronized Block
174. Synchronized Blocks Demo
175. InterThread Communication
176. InterThread Communication in Action
177. Multi-Threading
178. Disadvantages of traditional thread creation
179. Executor Framework Introduction
180. Implement CheckProcessorTask
181. Create a pool of threads

182. Callable and Future

183. Implement Callable

184. Use Future and read the result

185. Introduction to Garbage Collection

186. Basic GC Demo

187. Pushing the JVM Memory Limits

188. Requesting for Garbage Collection

189. Garbage Collection

190. Static Inner Classes With Static Members

191. Static Inner Classes With Non Static Members

192. Non Static Inner Classes

193. Accessing Outer Class Members

194. Local Inner Classes

195. Anonymous Inner Classes

196. Anonymous Connection Class

197. Anonymous Runnable Class

198. Inner Classes

199. Strings Introduction

200. Different ways to create a String

201. Strings and Wrapper Types

202. String Pooling

203. String Pooling In Action

204. Immutability in Action

205. Immutable Values

206. String Comparison

207. Object Comparison

208. String Methods

209. More String Methods

210. StringBuffer and StringBuilder

211. String Handling

212. IO Streams Introduction

244. Set Introduction

245. Random class

246. Using HashSet

247. Different Set Classes

248. Iterator

249. TreeSet of Strings

250. TreeSet of StringBuffers

251. ListIterator

252. Comparable and Comparator

253. Create a StringBuffer Comparator

254. Sort Strings by Length

255. Sorting Objects

256. Create a Object Comparator

257. Map Introduction

258. HashMap Demo

259. LinkedHashMap Demo

260. IdentityHashMap Demo

261. WeakHashMap Demo

262. Queue Introduction

263. PriorityQueue Introduction

264. PriorityQueue In Action

265. NavigableSet Introduction

266. NavigableSet In Action

267. Navigable Map

268. Arrays and Collections Classes

269. Collections Sort

270. Using Custom Comparator

271. Binary Search

272. Reversing a List

273. Arrays sort()

274. Arrays sort using custom comparator

275. Arrays Binary Search

276. Array to List conversion

277. Generics

278. Generic class structure

279. Create your own Generic Class

280. Restricting Generic Type Parameters

281. Using multiple restrictions

282. Using Generic Method Parameters and Wild Cards

283. Wildcard and extends

284. Wildcard and super

285. Method level generic type parameters

286. Type Erasure

287. Collections with generics

288. Enum Introduction

289. Using a Enum

290. Values and Ordinal methods

291. Defining and using fields

292. Enums

293. Introduction to new features of JDK 1.8

294. Lambda Expressions

295. Functional Interfaces

296. Create a functional Interface

297. Create your first Lambda

298. Addition using Lambdas

299. Lambdas using Runnable Interface

300. Lambdas and anonymous classes

301. Default methods in interfaces

302. Diamond Problem and Default Methods

303. Predicates

304. Predicate handson

305. String Predicate

306. Passing Predicate to a method

307. Predicate Joins

308. Predicate Joins in Action

309. Functions

310. Function Hands On

311. Method Referencing using :: Operator

312. Method Referencing in action

313. Referencing an instance method

314. Referencing a Constructor

315. Streams Introduction

316. Filter Even Numbers Using Streams

317. Convert Strings to Lower Case using streams

318. Other Methods on the Stream

319. What is a Virtual Machine?

320. Components of a JVM

321. How Class Loaders Work

322. Types of class loaders

323. Dynamic Class Loading In Action

324. Class is loaded only once

325. Display the class loaders

326. Class Loading Sub System

327. Linking

328. Initialization

329. Method Area

330. Stack Area

331. Heap Area

332. PC Registers Area

333. Native Method Stack Area

334. Introduction to internationalization

335. Locale

336. Locale hands on

337. NumberFormat

338. NumberFormat integers and fractions

339. DateFormat

340. DateFormat hands on

341. DateFormat Time instance

342. SimpleDateFormat Class

343. String to Date

344. Including Time

345. Introduction to annotations

346. Using @Deprecated

347. Annotations

348. Using @Override

349. Using @SuppressWarnings

350. @SuppressWarnings and Generic Types

351. Create User Defined Annotation

352. Use your annotation

353. Examine the inbuilt annotations

354. Introduction

355. 02:30

356. Key Reflection API Classes

357. Load the Class

358. List the constructors fields and methods

359. Create an Object

360. Invoke the Parameterized Constructor

361. Invoke the Getter

362. Invoke the Setter

363. Summary so far

364. Reflection

365. Modifying private fields

366. Accessing Annotations

367. Accessing fields on annotation

368. Components To Compile and Run a Java Program

369. Constructors

370. Overloading vs Overriding

371. Final Finally and Finalize

372. Generics and Type Erasure

373. == vs equals()

374. Java Class Loaders

375. serialVersionUID

376. Introduction to new features of JDK 1.9

377. Software Setup

378. Private Methods in interfaces

379. Static Private Methods

380. Improved Try with resource blocks

381. Immutable Collections

382. @SafeVarargs Enhancements

383. New Methods in streaming API

384. JSHELL Introduction

385. JShell in Action

386. More JShell

387. Modules Introduction

388. Create Modules

389. Use one module inside another

390. Packages are mandatory

391. Transitive Dependencies

392. Using Static and Cyclic Dependencies

393. Qualified Exports

394. Aggregator and Package Resolution Assignment

395. Introduction to JDK 1.10

396. Using var to declare variables

397. Uses of var

398. var restrictions

399. Collectors API updates

400. Assignment

401. Introduction to JDK 1.11

402. String API Updates

403. Files API Updates

404. isEmpty on Optional

405. Removals

Advance Java or J2ee

Part 1: Developing web application using JDBC , Servlets and JSP with MYSQL data base. Course would include several mini projects development for practical demonstration

1. Installing Tomcat

2. Configuring Tomcat in Eclipse

3. Downloading MySql and MySql workbench

4. Using MySql Workbench

5. Web Application Basics

6. Static vs Dynamic Web Applications

7. Server Side Programming

8. Introduction to servlets

9. Servlet Life Cycle Methods and Phases

10. Web Application Folder Structure

11. Servlets Introduction

12. Servlet Annotations

13. Hello World Servlet Project Creation

14. Code the Servlet

15. Register the Servlet

16. Application Flow

17. The Addition Use Case

18. Create the Servlet

19. Register the Servlet

20. Addition Application Flow

21. Introduction to JDBC

22. JDBC Architecture Introduction

23. JDBC API

24. JDBC Drivers

25. JDBC Client

26. DriverManager

27. JDBC Architecture Summary

28. Create the Account Table

29. Steps to perform CRUD Operations

30. JDBC Project Creation

31. Configure the Driver Jar

32. Connect to the Database

33. Connection String

44. JDBC Statement

45. Create an Account

46. Update the balance

47. Delete the Account

48. JDBC Steps To Read Data

49. JDBC ResultSet

50. Reading the Accounts

51. Navigating the ResultSet

52. Clean up JDBC Resources

53. The Service Provider Mechanism

54 Introduction to building dynamic web application

55. GET VS POST

56. User Application Use Case

57. Adding the mysql driver jar

58. Create the User Table

59. Create the User Servlet

60. Implement the init and destroy methods

61. Load the mysql jdbc driver class

62. Implement the doPost method

63. Create User Update Servlet

64. Create Static HTML

65. Create Dynamic HTML

66. Read User Servlet Flow

67. Servlet Init Params Introduction

68. Configuring Init Params using Annotations

69. Use annotated init params

70. Annotated Init Params in action

71. Configure using web xml

72. Use init params from web xml

73. Uses of ServletContext

74. Context Parameters

75. Configuring Context Parameters

76. Using Context Parameters in the code

77. Context Parameters in Action

78. PreparedStatement

79. Create the Product Table

80. Create the Project

81. Implement the init and destroy methods

82. Implement the doPost Method

83. Inter Servlet Communication

84. Usecase

85. Project Creation

86. Create the Login Servlet

87. Handle Login Success

88. Handle Login Failure

89. Create the Home Servlet

90. Application in action

91. Pre-Initialization

92. Preinitialization using annotations

93. Preinitialization using web.xml

94. Listeners

95. Create the Listener

96. Create the Servlet

97. Create the Filter

98. Introduction to sessions

99. HTTPs Statelessness

100.Steps for Session Management

101. Session Management Usecase

102. Create Source Servlet

103. How getSession works

104. What is Session Tracking

105. Create Target Servlet

106. Configure the Servlets

107. Application Flow

108. Ending a Session

109. Using Cookies

110. JSP Introduction

111. JSP Elements

112. JSP Life Cycle Methods and Phases

113. Hello JSP World

114. Implicit Objects

115. JSP Scripting Elements

116. Sum Of Two Numbers

117. JSP Directives

118. Account Creation Project Setup

119. Override jspInit and jspDestroy

120. Code the Insert Logic

121. Application in action

122. Exception Handling

123. Reading Cookies

124. Adding Cookies

125. URL Rewriting

126. JSP Actions

127. JSP UseBean Project Creation

128. Create the JSP

129. Application Flow

130. MVC Introduction

131. MVC Usecase

132. Create the Model

133. Create the Controller

134. MVC Final Flow

135. Introduction

136. Custom Tags Introduction

137. Project Creation

138. Create the tag handler class

139. Code the doStartTag method

140. Create a tag lib descriptor

141. Use the Tag

142. Custom Tag In Action

143. JSTL Introduction

144. Using JSTL Tags

145. c:set and c:remove

146. c:if

147. c:choose

148. c:forEach

149. Formatting Library

150. Format Numbers

151. Parse Date

152. Configure Users and Role

153. Configure Basic Authentication

154. Configure Form Based Authentication

155. Connection Pooling

156. Configuring a Connection Pool

157. Acquiring a Connection

158. JDBC Batch Updates

159. JDBC Batching - Project Setup

160. JDBC Batching - Code

161. JDBC Batching - In Action

162. ResultSetMetaData

163. ResultsetMetaData in Action

164. Transaction Management Basics

165. Transaction Management Usecase

166. Transaction Management in Action

HIBERNATE FRAME WORK

1. Software setup

2. Setting up H2 database

3. Creating the Maven project in Eclipse with Hibernate ORM dependencies

4. Creating the Category entity class

5. Creating the HibernateUtil class for getting a SessionFactory

6. Persisting a Category instance into RDBMS

7. Retrieving the Category instance from the RDBMS

8. Updating a Category description

9. Deleting a Category entity

10. Many-to-one association between Product and Category entities

11. One-to-many association between Category and Product entities

12. One-to-one association between Category and Product entities

13. Many-to-many association between Category and Product entities

14. Get all categories

15. Get products by price range

16. Pagination

17. Projection

18. Aggregate functions

19. Bulk update

SPRING BOOT , WEB SERVICES, ANGULAR, REACT WITH MICRO SERVICES

LEARN END TO END ENTERPRISE APPLICATION DEVELOPMENT

1. Introduction

Section 2: Quick Overview of Modern JavaScript,

1. Type Script and Angular

2. Getting Started with the Course

3. Understanding Full Stack Application Architecture

4. Quick Overview of Modern JavaScript and Type Script

5. Installing Angular CLI - Awesome Tool to create Angular Projects

6. Creating and Launching Angular Application with Angular CLI

Section 3: Getting Hands on With Angular

7. Importing Angular App into Visual Studio Code

8. Exploring Angular CLI Commands - test, lint, e2e, serve, build

9. Exploring Angular CLI Project Structure

10. Introduction to Angular Components -Basics

11. Introduction to Angular Components - AppComponent

12. Generating Welcome Component with ng generate

13. Language Variations With an Example - Java, JavaScript and Type Script

14. Generating and Setting up Login Component

15. Understanding Event Binding - Adding click event on Login Page

16. Using ngModel with 2 Way Data Binding in Login Page

17. Quick Review of Data Binding Approaches

18. Adding Hardcoded Authentication to Logic Component - ngIf directive

19. Implementing Routes for Login, Welcome and Error Components

20. Implementing Routing from Login to Welcome Component

21. Adding Route Parameter for Welcome Component

22. Create List Todos Component with ng generate

23. Create a Link to Todos in Welcome Component

24. Best Practice - Create a Todo Class

25. Quick Introduction to Angular Modules

26. Understanding Bootstrapping of Angular App with Root Module and Component

27. Quick Review - Angular Modules and Components

28. Overview of Next Few Steps - Bootstrap, Menu, Footer and Refactoring

29. Adding Bootstrap Framework and Creating Components for Menu and Footer

30. Using Bootstrap to Create a Menu with Navigation Links

31. Styling Footer and Other Components with CSS and Bootstrap

32. Good Practice - Use RouterLink instead of href for Routes

33. Creating an Independent Authentication Service Component

34. Using Session Storage to Store User Authentication Token

35. Enabling Menu Links Based on User Authentication Token

36. Implementing Logout to remove User Authentication Token

37. Securing Components using Route Guards

38. Quick Review - Authentication Service, Dependency Injection and Route

Section 4: Introduction to Web Services and REST

39. Portals where web services are used

40. What is a Web Service?

41. Important How Questions related to Web Services

42. Web Services - Key Terminology

43. Introduction to RESTful Web Services

Section 5: Getting Up and Running with REST and Spring Boot

44. Initializing a RESTful Services Project with Spring Boot

45. Creating a Hello World Service

46. Enhancing the Hello World Service to return a Bean

47. Quick Review of Spring Boot Auto Configuration and Dispatcher Servlet

48. Enhancing the Hello World Service with a Path Variable

Section 6: Connecting Angular Frontend to Spring Boot Restful Services

49. Connecting Angular Frontend with Restful API - 1 - Creating Data Service

50. Connecting Angular Frontend with Restful API - 2 - HttpClientModule an

51. Connecting Angular Frontend with Restful API - 3 - Understanding Observable

52. Connecting Angular Frontend with Restful API - 4 - Understanding Subscribe

53. Connecting Angular Frontend with Restful API - 5 - Handling Error Response

54. Calling Welcome HTTP Service with Path Variables

55. Designing RESTful Services for Todo Resource

56. Creating REST API for retrieving Todo List

57. Connecting Angular Frontend with Todo List RESTful Service

58. Creating REST API to delete a Todo - 1 - Create DELETE Request Method

59. Creating REST API to delete a Todo - 2 - Execute DELETE Request Method

60. Adding Delete Todo Feature to Angular Frontend

61. Creating Todo Component and Handle Routing

62. Designing Todo Page with Bootstrap Framework

63. Creating Retrieve Tod0 Service and Connect Angular Frontend

64. Improve Todo Page Appearance

65. Creating REST API for Updating Todo - PUT Request Method

66. Part 1 - Creating REST API for Creating a Todo - POST Request Method

67. Part 22 - RESTful Web Services - Best Practices

68. Implementing Update Todo Feature in Angular Frontend

69. Implementing New Todo Feature in Angular Frontend

70. Improving Todo Form - Validation and Form Submit on Enter - ngSubmit

71. Enhancing Validation Messages on Todo Page

Section 7: Implementing Basic Authentication with Spring Boot and Spring Security

72. Overview of Security with Basic Auth and JWT

73. Setting up Spring Security

74. Configure standard userid and password

75. Enhancing Angular Welcome Data Service to use Basic Auth

76. Configure Spring Security to disable CSRF and enable OPTION Requests

77. Creating Angular HttpInterceptor to add Basic Auth Header

78. Configure HttpInterceptor as Provider in App Module

79. Create Basic Authentication RESTful Service in Spring Boot

80. Create Angular Basic Authentication Service

81. Connect Login Page to Basic Authentication Service - Part 1

82. Connect Login Page to Basic Authentication Service - Part 2

83. Refactoring Angular Basic Authentication Service

84. Refactoring HttpInterceptor to use Basic Authentication Token

85. Best Practice - Use Constants for URLs and Tokens

Section 8: Connecting Spring Security and Spring

Boot with JWT Framework

86. Introduction to JWT

87. Importing JWT Framework into Eclipse

88. Quick Tip - Resolving JWT Compilation Errors

89. Executing JWT Resources - Get Token and Refresh Token

90. Understanding JWT Spring Security Framework Setup

91. Creating a New User with Encoded Password

92. Using JWT Token in Angular Frontend

Section 9: Connecting Spring Boot RESTful API With JPA and Hibernate

93. Setting up Todo Entity and Populating Data

94. Connecting GET REST APIs to JPA Repository

95. Connecting POST, PUT and DELETE REST APIs to JPA Repository

Data Structures:

Section 1: Introduction

1. Introduction to Data Structures

2. Introduction to Algorithms

Section 2: Arrays and Big-O Notation

3. Introduction to Arrays

4. Big-O Notation

5. A Quick Review of Arrays in Java

6. Arrays in Memory

7. Big-O Values for Array Operations

Section 3: Sort Algorithms

8. Introduction to Sort Algorithms

9. Bubble Sort (Theory)

10. Bubble Sort (Implementation)

11. Stable vs. Unstable Sort Algorithms

12. Selection Sort (Theory)

13. Selection Sort (Implementation)

14. Insertion Sort (Theory)

15. Insertion Sort (Implementation)

16. Shell Sort (Theory)

17. Shell Sort (Implementation)

18. Recursion

19. Merge Sort (Theory)

20. Merge Sort (Implementation)

Section 4: Lists

21. Introduction to Lists

22. Abstract Data Types

23. Array Lists

24. Vectors

25. Singly Linked Lists (Theory)

26. Singly Linked Lists (Implementation)

27. Doubly Linked Lists (Theory)

28. Doubly Linked Lists (Implementation)

29. The JDK LinkedList Class

30. Linked Lists Challenge #1

31. Linked Lists Challenge #1 Solution

32. Linked Lists Challenge #2

Section 5: Stacks

33. Introduction to Stacks

34. Stacks (Theory)

35. Stacks Implementation (Array)

36. Stacks Implementation (Linked List)

37. Stacks Challenge

38. Stacks Challenge Solution

39. Linked Lists Challenge #2 Solution

40. Quick Sort (Theory)

41. Quick Sort (Implementation)

42. Counting Sort (Theory)

43. Counting Sort (Implementation)

44. Radix Sort (Theory)

45. Stable Counting Sort (Theory)

46. Radix Sort (Implementation)

47. Sorting Arrays Using the JDK

48. Sort Algorithms Challenge #1

49. Sort Algorithms Challenge #1 Solution

Section 6: Queues

50. Introduction to Queues

51. Queues (Theory)

52. Queues (Array Implementation)

53. Circular Queue Implementation (Part One)

54. Circular Queue Implementation (Part Two)

55. Queues and the JDK

56. Queues Challenge

57. Queues Challenge Solution

58. Sort Algorithms Challenge #2

59. Sort Algorithms Challenge #2 Solution

60. Sort Algorithms Challenge #3

61. Sort Algorithms Challenge #3 Solution

Section 7: Hashtables

62. Introduction to Hashtables

63. Hashtables (Theory)

64. Hashtables (Array Implementation)

65. Linear Probing

66. Linear Probing - Removing Items

67. Linear Probing - Rehashing

68. Chaining

69. Hashtables and the JDK

70. Bucket Sort (Theory)

71. Bucket Sort (Implementation)

72. Hashtables Challenge #1

73. Hashtables Challenge #1 Solution

74. Hashtables Challenge #2

75. Hashtables Challenge #2 Solution

Section 8: Search Algorithms

76. Introduction to Search Algorithms

77. Linear Search Algorithm

78. Binary Search Algorithm

79. Binary Search (Implementation)

Section 9: Trees

80. Introduction to Trees

81. Trees (Theory)

82. Binary Search Trees (Theory)

83. Binary Search Trees (Insertion)

84. Binary Search Trees (Traversal)

85. Binary Search Trees (Get, Min, Max)

86. Binary Search Trees (Delete Cases 1 and 2)

87. Binary Search Trees (Implement Cases 1 and 2)

88. Binary Search Trees (Delete Case 3)

89. Binary Seach Trees (Implement Case 3)

90. Trees and the JDK

91. Binary Search Trees Challenge #1

Section 10: Heaps

92. Introduction to Heaps

93. Heaps (Theory)

94. Storing Heaps as Arrays

95. Heaps (Insert)

96. Heaps (Delete Theory)

97. Heaps (Delete)

98. Heaps (Peek)

99. Priority Queues

100. Heapsort (Theory)

101. Heapsort (Implementation)

102. Binary Search Trees Challenge #1 Solution

103. Binary Search Trees Challenge #2

104. Sets

Resume Building

1. Resume designing with cover letter content creation

2. Resume Uploading on job portals by optimizing it