Please find below details on requirement-

**1.High level topics/ Table of content to be covered-**

|  |  |
| --- | --- |
| **Module Name** | **Topics** |
| Introduction to Java | Introduction to Java Bytecode,Class Files,Compilation Process,Data types and Operations, if conditions,Loops – for, while and do while |
| Data Handling and Functions | Arrays - Single Dimensional and, Multidimensional, arrays,Functions, Function with Arguments Function Overloading,The concept of Static Polymorphism String Handling - String, StringBuffer Classes |
| Object Oriented Programming in Java | OOPS in Java: Concept of Object, Orientation Attributes, and Methods |
| Object Oriented Programming in Java - 2 | Classes and Objects Methods and Constructors – Default,Constructors and Constructors with Arguments,Polymorphism, Inheritance ,Abstract, Final, Static |
| Packages | Packages and Interfaces, Access Specifiers: Public, Private, Protected and Package, |
| Multi-Threading and Collections | Exception Handling: Try, Catch, Finally, Throw and Throws,Multi- Threading: Runnable Interface, Extending a Thread Class, Synchronization in Threads, Executor Framework, Wrapper Classes and Inner Classes: Integer, Character, Boolean, Float etc. |
| Collections - 2 | Java. Lang, Java.io, Java.util Collections: Array List. Vector, HashSet, Tree Set, HashMap, Hash Table |
| JDBC | Introduction to SQL: Connect, Insert, Update, Delete, Select,Introduction to JDBC and Architecture of JDBC Types of Drivers: Type 1/2/3/4 drivers Insert/Update/Delete/Select Operations using JDBC Batch processing Transaction Management: Commit and Rollback |
| Hibernate | Introduction to Hibernate Architecture of Hibernate Database Operations: Insert/Update/Delete/Sheet Inheritance, Collections, HQL and Restrictions Caching in Hibernate |
| Servlets | Introduction to Web Technologies, Type of Servlets: Generic and Http Servlet, Request Dispatchers: Forward and include 4 types of Session Tracking and Filters,jsp tags (Scripts, declarative, expression) |
| Maven | Maven (Local, Remote and Central Repo) Setup Maven, Manage Repositories and Dependency Management, Building and deploying project, create package (jar or war), Integrate Maven with Eclipse Manage testing and deployment options, Maven Goals, Maven Plugins |
| Introduction to Spring | Introduction to Spring Framework, Why Spring, Java Frameworks, Spring Framework, architecture, Object Coupling Problem, Loose coupling and Tight coupling, Dependency Injection and its types, Bean Life cycle, Bean scopes (singleton and prototype) |
| Spring Configurations | What is Autowiring, Different types of autowiring, Implementing autowiring in different ways, Spring bean definition inheritance, Specifying default values using @value annotation, Autowiring bean using @Autowired, Resolving bean using @Qualifier, Defining a component using @Component, Using @PostConstruct and @PreDestroy, Automatic Bean discovery in a Spring application, Using Spring Expression Language(SpEL) |
| Aspect Oriented Programming and DAO | What is Aspect Oriented Programming (AOP),AOP terminologies, Implementing, MethodBeforeAdvice, Implementing AfterReturningAdvice, Implementing MethodInterceptor Implementing ThrowsAdvice, What is AspectJ, AspectJ provided annotations like @Aspect,@Before etc., Data Access Object (DAO)pattern, Using property files, |
| Spring and Design Patterns | Design Patterns: DAO, DTO, MVC, Intercepting filters Front Controller, Business Delegate, Spring MVC Architecture, MVC Components, Setup Spring MVC Application, Writing a Spring MVC application, Exception Handling, Apache Tiles. |
| Data Access | Using Jdbc Template, Binding variables in database query, Mapping database rows to Java class, Jdbc Batch Processing |
| Data Access - 2 | What is Hibernate, Working With Hibernate, Integrating Hibernate with, Spring, Database Transaction, Spring support for Database transactions |
| Web Services and Project | RESTfulWeb Service JAX-WS Implementation |
| Introduction to Design Patterns | Inheritance, Polymorphism, Interfaces, Different Member Field Attributes, Understand Design Patterns, Design Patterns using well defined UML Diagrams, Importance of Design Patterns, Introduction to different Classes of Design Patterns, Implementing DAO and Factory Pattern. |
| Creational Design Patterns | Abstract Factory, Builder, Factory, Prototype, Singleton, Object Pool. |
| Structural Design Patterns | Adapter, Bridge, Composite, Decorator, Facade, Flyweight, MVC, Front Controller, Module, Proxy |
| Behavioural Patterns | Chain of Responsibility, Command, Iterator, Mediator, Observer, Strategy, Template Method, Visitor. |
| Spring Batch | What is batch processing? Why build batch processes? Batch processing with Spring Batch stereotypes Batch processing in the cloud Integration with common technologies |
| Integrating Struts 2 | Understand Struts 2 and JSF,  Integration Spring with Struts, Developing a Struts 2 application |
| JSF with Spring and Spring Web Flow | Integrating JSF with Spring, Developing a JSF application,  Spring Web Flow. |
| Spring Security | Spring features for Securing applications, Implementing Remember me feature, Storing Encrypted Passwords, Restricting the number of login attempts, Logging with Log4j, Writing test cases with JUnit |
| Spring Integration | Introduction to messages, channels, and Endpoints, Receiving Messages, Common Endpoints, Endpoint API, Message Flow components, Introduction to Adapters. Integration with Big data. |
| Git and GitHub | Source Control, Git, Key Git Terminology, Basic Git Commands, Branching and Merging |
| Introduction to Microservices | Monolithic Architecture, Distributed Architecture, Service oriented Architecture, Microservice and API Ecosystem, Microservices in nutshell, Point of considerations, SOA vs. Microservice, Microservice & API |
| Spring Boot - Introduction and Installing Basic Tools | Use Spring Initializr to download a skeleton Spring boot project template, then configure and use eclipse IDE to import the skeleton project |
| Spring Boot - Bootstrapping an application | Understand the structure of the skeleton project and use of Dev-Tools for faster application development |
| Spring Boot - Accessing Data with Spring Boot and H2 | Understand and implement the concept of embedded DB, access H2 Console UI, query data using JPQL NoSQL(Firebase) |
| Spring Boot - Configuring a Spring MVC Application with Spring Boot | Create JPA entities, use Lombok Framework to generate getters, setters, toString, equals and hashcode methods |
| Spring Boot - Building a RESTful Web Application with Spring Boot | Understand what are RESTful API's, Create RESTful API's |
| Spring Boot - Building a RESTful Web Application with Spring Boot - 2 | Test Restful API's |
| API Testing with SOAP UI | Introduction to SoapUI, Installing SoapUI, The SoapUI Interface, Your First REST Test, Your First SOAP Test, Mocking |
| Spring Boot - Using Swagger(OpenAPI) and API documentation | Understand what is API Documentation and then configure and use Swagger UI for rapid API's testing |
| Spring Boot - Enabling Actuators, Metrics, and Health Indicators | Using Spring Boot features |
| Spring Boot - Authentication and Authorization | Authentication using Basic Auth, Oauth2 and JWT Firebase Authentication |
| Spring Boot - Testing with Spring Boot | Unit Testing, Integration Testing, Contract Testing and End to End Testing |
| Testing with JUnit | Introduction to JUnit, Introduction to Mockito and writing a test case Using Mockito Annotations - @Mock, @InjectMocks, @RunWith(MockitoJUnitRunner.class), Creating the Project with Spring Initializr, Starter Projects in pom.xml, Add the code under Test Unit Testing with Mockito using MockitoRunner,Unit Test launching the complete Spring Context using @MockBean |
| Spring Cloud Concepts | Spring Cloud concepts |
| Cloud Native Concepts-Containerization | Docker, Kubernetes and Cloud Native concepts |
| Introduction to Microservices | Introduction to Microservices, Monolithic Architecture SOA Architecture,Key benefits of Microservices Challenges in Microservices Comparisons between Monolithic, SOA and Microservices Microservices: Process and Organization Use Case #1 FTGO |
| Microservices Design and Architecture | Introducing Microservices Architecture,Microservices Design Patterns Use case: Apollo Store, Decomposition Strategies Obstacles in Decomposition,Inter-process communication Partial failure in communication,Service Discovery Transaction management  Saga, Partial, Full fail over using Circuit breaker,  Event Driven Gate way Blue/Green Deployment Strategy etc |
| Introduction to Springboot and Spring framework - I | Introduction to Springboot Comparisons between Spring, Springboot Working of Spring in Eclipse/any other IDE of your choice Necessary files in Springboot   -Components   -POM   -Servers Annotations Spring boot database – H2 and JPA |
| Springboot and Spring framework - II | Spring boot AOP, Springboot caching, Sprinboot database - MySQL Springboot RESTful Web Services,Swagger documentation format |
| Microservices with Springboot and Cloud | Introduction to Spring cloud, Difference between Springboot and Spring cloud, Spring Cloud features, Spring Cloud and Netflix Eureka Registration and Discovery,Spring Load Balancer Introducing Spring Cloud Gateway |
| Microservices Security | Introduction to Spring Security, Microservices Security Principles Introduction to OAuth 2.0 |
| Microservices Security | Authentication and Authorization, Principal Security Springboot Security |
| Getting Started with Angular | Building Blocks of Web Application Development Web Application Architecture,  Introduction to Angular Comparison between front-end tools, Angular Architecture Building blocks of Angular,  Angular Installation Angular CLI, Angular CLI commands, Angular Modules Understanding files in Angular |
| Angular Components and Data Binding | Working of Angular Applications,  Angular App Bootstrapping Angular Components,  Creating A Component Through Angular CLI Ways to specify selectors,  Template and styles, Installing bootstrap to design application, Data Binding,Types of Data Binding Component Interaction using @Input and @Output decorator Angular Animations Component Life-cycle Hooks |
| Directives and Pipes in Angular | Understanding Angular Directives,  @Component Directive Structural Directives, Attribute Directives, Custom Directives Pipes, Built-in Pipes, Chaining pipes, Custom pipes PipeTransform Interface & Transform Function Pure and Impure pipes |
| Angular Services and Dependency Injection | Angular service, Need for a service Dependency Injection Creating a service Hierarchical Injector, Injecting A Service into Another Service,  Observables, RxJS Library, Angular’s Interaction with Backend, Parts of an Http Request, HttpClient |
| RxJS and HTTPClient | RxJS Library, Angular’s Interaction with Backend , Parts of an Http Request, HttpClient |
| Angular Routes and Navigation | Angular Router, Setting Up Routes, Adding Routes Using RouterLink Wildcard and Redirecting Routes,  Adding Navigation Programmatically, Passing Route Parameters |
| Angular Routes and Navigation - 2 | Extracting Parameters Using ActivatedRoute, Optional Route Parameters, Child Routes, Route Guards, Location Strategies |
| Handling Forms in Angular | Angular forms, Types of forms, Underlying building blocks of the form model, Template-driven vs Reactive forms, Template-driven forms Reactive Forms,  Dynamically adding data to a form |
| Validating Angular Forms | What is Form Validation?  Types of Form Validation, Built-in Validators, Form control’s status and validity, Form Validation methods, CSS classes for Form control, Custom validators in Template Driven Forms |
| Authentication with JWT and Security | What is Authentication?, Authentication and authorization Types of Authentication, Where to store tokens? JSON Web Tokens (JWT), Authentication in Angular application Security threats in web application |
| Testing and Application Deployment in Angular | Testing,  Why should we perform testing? Types of testing Testing Angular application using Jasmine and Karma Maintaining application code using Git,  Version control system Why should we use Git?  Git file workflow, Running application on production server: Nginx,  Architecture of Nginx How to configure Nginx?,  Deployment of an application using Docker Problems before containers,  How containers solve the problems What is Docker?,  Docker file, Docker image , Docker containers Docker hub,  Basic Docker commands, ANGULAR PROJECT BRIEFING |
| Overview of DevOps | Introduction to DevOps, Benefits of working in a DevOps environment DevOps Lifecycle, DevOps Stages, DevOps Delivery Pipeline |
| Version Control with Git | Version Control Preview, Git Introduction Preview Git Installation, Commonly used commands in Git Working with Remote repository |
| Git, Jenkins & Maven Integration | Branching and merging in Git Preview, Merge Conflicts Stashing, Rebasing, Reverting and Resetting Git Workflows,Introduction to Maven, Maven Architecture Introduction to Continuous Integration, Introduction to Jenkins |
| Continuous Integration using Jenkins | Jenkins Architecture, Plugin Management in Jenkins Preview Jenkins Security Management, Notification in Jenkins Jenkins Master-slave architecture, Jenkins Delivery Pipeline Jenkins Declarative pipeline |
| Configuration Management Using Ansible | Introduction to Configuration Management, Infrastucture as Code Introduction to Ansible, Ansible Architecture, Inventory Management Ansible Modules, AD-HOC Commands ,Ansible Playbooks Preview Ansible Roles |
| Containerization using Docker Part – I | Containerization Preview, Namespaces, Docker Docker Architecture, Container Lifecycle, Docker CLI, Port Binding |
| Containerization using Docker Part – II | Detached and Foreground Mode, Dockerfile, Dockerfile Instructions Docker Image, Docker Registry, Container Storage Volumes, Docker Compose, Docker Swarm Introduction to Docker Images, Docker containers, Docker Images, Docker Repositories Managing containers for Microservices, Monitoring with Prometheus and Grafana |
| Orchestration using Kubernetes Part - I | Introduction to Container Orchestration, Kubernetes Core Concepts Understanding Pods, ReplicaSet and Replication Controller Deployments Preview, DaemonSets, Rolling Updates and Rollbacks Scaling Application, Services, Persistent Storage in Kubernetes Primitives for PersistentVolumeClaims, Secrets and ConfigMaps Headless Services, StatefulSets, Helm Charts |
| Introduction to Machine Data & Splunk Basics | What is Machine Data & its challenges?, Need for Splunk and its features, Splunk Products and their Use-Case, Download and Install Splunk, Splunk Components: Search Head, Indexer, Forwarder, Deployment Server, & License Master, Splunk Architecture Splunk Licensing options |
| User Management & Splunk Configuration Files | Introduction to Authentication techniques, User Creation and Management, Splunk Admin Role & Responsibilities Indexes,Data Ageing,Introduction to Splunk configuration files (7) Managing the. conf files |
| Data Ingestion, Splunk Search, and Reporting Commands | Learn the various data onboarding techniques: - Via flat files, Via UF (Universal Forwarder) Implement Basic search commands in Splunk: - Fields, Table, Sort, Rename, Search Understand the use of time ranges while searching Learn Reporting & Transforming commands in Splunk: - Top, Rare, Stats, Chart, Timechart, Dedup, Rex |
| Splunk Alerts, Visualizations, Reports, & Dashboards | Create Alerts triggered on certain conditions Different Splunk Visualizations Create Reports with search results Create Dashboards with different Charts and other visualizations Set permissions for Reports and Dashboard Create Reports and schedule them using cron schedule Share Dashboard with other teams |
| Java Streams | Creation of Streams Use cases for Streams Operations of Streams |
| Java Generics | Why Generics? Type of Generics. Advantages of Generics |