**ASDE Training -TOC - 2023**

**Day 1**

* Introduction to Object Oriented Analysis & Design (OOAD)
* Classes and Objects
* Object oriented concepts
* Fundamentals of UML
* Introduction to UML
* Goals of UML
* Use Case Modeling
* UML Diagrams
* Class and Object Diagrams
* Interaction Diagrams - Sequence and Collaboration
* Introduction to Maven & POM file
* Maven Lifecycle & Repositories
* Introduction to gradle (Overview)
* Differenece between maven and gradle

**Day 2**

* Our First Repository – Git
* Branching and Merging
* Introducing to GitHub –
  + Creating account and cloning all codes to github/bitbucket
* Why Jenkins?
* What is Continuous Integration and Continuous Deployment
* Jenkins installation and Configuration
* Running Jenkins as windows service
* Deploying Jenkins in Tomcat Server
* Jenkins Global Configuration
* Building your first Hello world job
* Jenkins Plugin management
* Understanding the Jenkins workspace
* Build
* pre build and post build job

**Day 3**

* Introduction to NOSQL databases (MongoDB)
  + What is MongoDB?
  + NoSQL databases v/s Relational dbs
  + When to prefer MongoDB over other NoSQL dbs?
  + The current SQL/NoSQL landscape
  + Document-oriented vs. other types of storage
  + Mongo's features
  + Common use-cases
  + Introduction to JSON & BSON
* Documents and Collections
  + Creating documents
  + Managing documents in collections
  + Iterating over documents
* CRUD Operation using CLI and Mongo Compass
* Creating Indexes
* CAP Theorem

**Day 4**

* Java Collections and Generics
  + The Collections Framework and its API
  + Collections and Java Generics
  + Collection, Set, List, Map, Iterator
  + Auto boxing
  + Collections of Object (non-generic)
  + Using ArrayList, HashSet, and HashMap
  + Processing items with an Iterator
  + More about generics
  + Loosely Coupled Systems

**Day 5**

* Introduction to SQL (MySQL)
* Data types
* Introduction to DQL, DDL, DML, DCL
* CRUD operations with database
* INSERT/UPDATE/DELETE/RETRIEVE
* Using functions and ordering the result
* Database Access with JDBC
  + JDBC Overview
  + Steps to connect with Database
  + DriverManager,
  + Connection,
  + What is Statement, PreparedStatement, CallableStatement
  + ResultSet
* CRUD Operations using JDBC with MySQL DB

**Day 6**

* Lambda Expressions
* Interface Static Methods
* Default Methods
* Functional Interfaces
* Lambda Syntax
* Using Lambdas
* Function Descriptors
* Static & Instance Method References
* Constructor References

* Streams API
* What are Streams?
* Stream Advantages
* Stream Operations
* Streams vs. Collections
* Filtering and Mapping

**Day 7**

* Introduction to Java 11 & 17
* Writing your first Java 11 program
* Exploring strings new Features
* Sealed Classes
* Pattern Matching
* Documentation using Javadoc
* Working with Strings, Characters and Regular Expressions
* Error handling in Java 11
* NIO & NIO2

**Day 8**

* TDD – Test Driven Development (Overview)
* JUNIT (Junit5)
  + What is JUnit?
  + Features
  + JUnit test case
  + Getting started
  + Using JUnit
  + Test methods: testMethod or @Test method
  + Assertions in TestCase
  + Testing for exceptions
  + Testing isolations
  + Creating and organizing test fixtures with setUp/tearDown and @AfterAll@AfterEach, @BeforeAll, @BeforeEach
  + @ParameterizedTest
* MOCKITO
  + Mockito for mocking objects
  + Using Mockito
  + Limitations
  + Configuring the mock objects
  + Verify the calls on the mock objects
  + Stub/Mock/Spy/Test Double
* Static Code Analysis using Sonarqube
  + Analysing the Code using Sonarqube
  + Adding Sonar Lint in IDE
  + Code Smell
* Introduction to Logging
  + Using Log4J
  + Configuring SLF4J
  + Logging Levels (Info, Debug, Warn, Error)
  + Log message formats & Logging config
* Lombok
  + Lombok Annotations
  + @Data/@Getters/@Setters
  + @NoArgsConstructor/@AllArgsConstructor

**Day 9**

Reviewing Design Patterns & Principles in Java

* Importance of Design Patterns
* Different Types (Creational, Structural, Behavioral, Concurrency)
* Understanding SOLID Principles and need
* Clean Coding Practices
* Understand Your Code
* Comments are your buddies
* Done Repeat Yourself (DRY)
* Indent Your code
* Naming Conventions
* YAGNI (You aren’t Gonna Need It)

NFR (Non Functional Requirements)

* Availability
* Scalability
* Security
* Extendibility
* Flexibility

**Day 10**

**Types of Servers**

* Web Server
* Application Server
* Database Server

**Servlets**

* What are servlets (Html Code Embedded in JAVA)
* Coding issues, Types of Servlets [Generic & Http]
* ServletResponse, ServletRequest etc…
* Servlet life cycle (Overview)
* Passing parameters to a Servlet

**JSP**

* Why JSP Server Pages (Java code embedded in Html)
* JSP Access Model
* Syntax
  + Directives <%@ %>
  + Declarations <%! %>
  + Scriptlets <% %>
  + Expressions <%= %>
  + Action Tags
* JSTL – CRUD operation using JSTL

**Day 11**

* Introduction to Spring
* IOC: All Frameworks use the IoC principle
* Dependency Injection: Spring framework supports
* dependency injection
* Using Spring to configure an application
* simplifying application configuration
* Annotation-based dependency injection
* Spring configuration.
* Driving database transactions in a Spring environment

**Day 12**

* Spring Web
* Spring Web application
* Integrating Spring MVC in web application
* Building application with spring MCV
* Working with annotation and spring MVC
* Spring AOP
  + What is and Why AOP?
  + AOP concepts and terminology
  + @AspectJ support in Spring
  + Types of advice
  + Declaring advices
  + JointPoint

**Day 13**

* SPRING BOOT INTRODUCTION
  + Introduction to Spring Boot
  + Value Proposition of Spring Boot
  + High-level Spring Boot features
  + Creating a simple Boot application using Spring

Initializr web-site

* SPRING BOOT DEPENDENCIES, AUTO-CONFIGURATION AND RUNTIME
  + Dependency management using Spring Boot starters
  + How auto-configuration works
  + Configuration properties
  + Using CommandLineRunner
  + Using In-memory Database (h2)
* Introduction to MongoReposiotory
  + Performing CRUD Operations with MongoDB
* Testing the Endpoints using Postman
* API Documentation using Swagger

**Day 14**

Introduction to Spring Cloud, Circuit Breaker, Hystrix, Open Feign

**Hystrix**

* Knowing what is circuit breaking
* Setting up a micro service application
* Apply circuit breaker pattern

Eureka

* Locating services
* Load balancing
* Knowing failovers of middle tier
* Client-side balancing

**Day 15**

* Introduction To Web Development
  + How web applications work
  + An introduction to HTML and CSS
  + Tools for web development
  + How to view deployed web pages
  + Five critical web development issues
  + The HTML syntax
* Images
  + Basic skills for working with images
  + Advanced skills for working with images
  + Related skills for working with images
  + Using Media Queries
* Forms
  + How to use forms and controls
  + Other skills for working with forms
  + How to use the HTML5 features for data validation
  + How to use the HTML5 controls
  + A web page that uses HTML5 data validation
* Using CSS to format the elements of a web page
  + An introduction to CSS
  + Measurements and colors
  + Selectors
  + Cascading Style Sheets
  + How to work with text
  + A web page that uses external style sheets
  + Introduction to media queries

**Day 16**

* Introduction to JavaScript
  + Introduction to JavaScript language
  + Using Script tag
  + Inserting Java Script into tags
  + Linking Java Script from separate files
  + JavaScript expression and Operators
  + Defining and naming variables
  + Data Types
  + Expression
  + Working with es6 standards
  + Returning values from functions
* Intro to AJAX
* Functional Programming
* ES6 Features (call, Async, await)
* JQuery introduction

**Day 17**

* Introduction to NodeJS, NPM, Modules
* ReactJS Fundamentals
  + JavaScript MVC
  + View Technology
  + Introduction to ReactJS
  + React and SPA
  + The create-react-app CLI
  + Project structure
  + React Introduction
  + View Technology
  + React and SPA
  + The create-react-app CLI
  + Project structure
  + Understanding What React Is
  + What Problem React Solves
  + Virtual DOM Concept
  + Comparative Analysis
  + React Foundation
  + React Fitting Into Web Applications
  + Modern Web Architecture
  + Developer Setup
  + IDE Plugins
  + Chrome Plugins
  + Creating First React Project
  + Code Formatter Setup
  + Building Static Site Using React
  + JSX Introduction

**Day 18**

* States and Properties
* Stateless Components
* Stateful Components
* Styling Approach in React
* Component Lifecycle Events
* Event Categories
* Event Definition
* Mounting and Unmounting Events
* Components and Props
* State and Lifecycle
* Handling Events
* HOC (Higher Order Components)

**Day 19**

* React Forms, Hooks
* Error Boundaries
* React.lazy and Suspense for CSR/SSR
* Handling Events in React
* Reacting to User action in React
* Advanced React JS
  + Using pure functions as components
  + Custom middleware

**Day 20**

* Redux for state management
  + Introduction to Redux
  + Basic building blocks
    - Store
    - Actions, action types and action creators
    - Dispatch mechanism
  + Vanilla Redux for better understanding
* Redux with ReactJS
  + The application flow
  + Choosing the right model for store
  + Implementing Redux for a basic application
* React Routers
* Testing React Application

**Day 21**

* Introduction to Behavior-Driven Development
* Test-Driven Development as a Design Tool
* TDD & BDD in Real-World Setting, TDD Beyond
* Creating TestRunners to Execute Tests
* How to Write Feature Files with Gherkin Language
* Introduction to Cucumber
* More about e2e Testing
* Selenium (Overview)
* Creating Tags to on Feature File
* Features and Scenarios: Given-When-Then Outline
* Overview of Selenium Webdriver, pros and cons, Extracting Xpaths, CssSelectors
* Downloading and configuring Java WebDriver in eclipse, WebDriver Interface,
* Working with Firefox, Chrome browsers
* Identifying Web-Elements using id, name, linkname, class, xpath, tagname
* Handling Input box/buttons, list/selection/drop down boxes, radio buttons

**Day 22**

Publish/Subscribe Messaging

* Origin Story
* What Is KAFKA
* Why Kafka
* Messages & Batches
* Schema
* Topics & Partitions
* Kafka Client
* Brokers & Clusters
* Multiple Clusters
* Kafka Installation requirements
* Producer/Consumer Pattern (Pub/Sub model)

**Day 23**

Docker

* Introduction to Docker
* Docker Architecture
* The Docker Hub
* Docker Installation
* Creating Our First Image
* Working with Multiple Images

Containers

* Packaging A Customized Container
* Running Container Commands with Docker
* Exposing Our Container with Port Redirects
* Exercise: Exercise: Installation and Image Setup
* Exercise: Exercise: Creating Images from Containers
* Exercise: Exercise: Exposing Container Ports to the Host
* Docker Hub, Docker Compose

**Day 24**

Introduction to Kubernetes

* Introduction to container orchestration
* Installing Minikube
* Container Orchestration
* Cluster, Control Pane, Nodes & Pods
* Replication & Scaling
* deploying a spring boot application with k8s

**Day 25**

* Creating AWS Account, understanding ec2 instance
* Introduction to Amazon Simple Storage Service (S3)
  + S3 Regions, Buckets and Objects
  + Example: Create Bucket in S3
  + Example: Upload - Make Public - Rename - Delete File in S3
  + Example: Delete and Restore Files in S3
  + S3 REST API Overview
  + Durability, Availability and Data Consistency
  + S3 Access Control
  + Understanding NAT & Security Policies

**Additional Reads**

**Working with Netflix tools**

* Micro services spring cloud, feign(proxy)
* Building simple applications with Cloud
  + Initial values from .properties file
  + Knowing what is proxy object

**Over view of Sonar**

* Installation of SonarQube
* Installation of Code Analyser
* Sonar profiles
* Running Sonar for a project
* Configuring SonarQube for one Sample Java project
* Integrating SonarQube with the project

**TDD**

* Introduction to Unit test, unit test characteristics
* Test Driven Development – why, what and how
* TDD Cycle – Red, Green and Refactor
* Drive the development (design and implementation) of a User Story
* Learn concept of Emergent Design and why emergent design
* How TDD helps you design software
* Why testing first is better than testing after

Working with Spring Test Suite

**Introduction to Elements of JMeter Test Plan**

* Thread Group
* Controllers
* Samplers
* Logic Controllers
* Listeners
* Timers
* Assertions
* Configuration Elements
* Pre-Processor Elements
* Post-Processor Elements

**Dockerfile Directives**

* Dockerfile Directives: USER and RUN
* Dockerfile Directives: RUN Order of Execution
* Dockerfile Directives: ENV
* Dockerfile Directives: CMD vs. RUN
* Dockerfile Directives: ENTRYPOINT
* Dockerfile Directives: EXPOSE