**Backend**

**Duration: 112 Hours**

**OOPS Concept and Java (20 Hours)**

**Types Of Programming Languages**

* Unstructured Vs Structured Programming Languages
* Aspect Oriented Programming Languages
* Object Based Vs Object Oriented Programming Languages

**Object Oriented Features**

* Class Vs Object
* Encapsulation Vs Abstraction
* Inheritance
* Polymorphism
* Message Passing

**Containers or top most elements in java**

* Class Syntax
* Access Modifiers
* class, className, extends, implements keywords
* Possible Syntaxes of Classes
* Procedure to use classes in Java
* Internal flow in Class Utilization
* More than one class in Single Java Appl
* Concrete Methods Vs Abstract Methods
* Abstract Classes
* Interfaces Part

**Methods in java**

* Method Syntax Part - 1
* Method Signature and Prototype
* Mutator Methods Vs Accessor Methods
* Var-Arg Method

**Object creation process in java**

* Procedure To Create Objects
* hashCode() and toString() methods
* Immutable Objects Vs Mutable Objects
* User defined Immutable Class
* Object Vs Instance

**Constructors in java**

* Constructors : Introduction
* Default Constructor
* User Defined Constructors
* Constructors : Introduction
* Default Constructor
* User Defined Constructors
* Constructors : Introduction
* Default Constructor
* User Defined Constructors
* Constructor Overloading

**Instance context in java**

* Instance Variable
* Instance Methods
* Instance Block and Instance Flow Of Execution
* Instance Flow Of Execution

**‘This’ keyword in java**

* 'this' keyword : To refer current class Variables
* 'this' keyword : To refer current class Methods
* 'this' keyword : To refer current class Constructors
* 'this' keyword : To return current class Object

**‘static’ keyword in java**

* 'static' keyword: Static Variable
* 'static' keyword: Static Method
* 'static' keyword: Static Block
* 'static' keyword: Static Anonymous Inner class
* 'static' keyword: Static Import
* 'static' keyword: Static Context
* 'static' keyword: Static Context and Instance Context

**Class.forName() method internal functionality**

* Utilizations of Class.forName() and newInstance() methods

**Factory Methods and Singleton classes**

**final keyword and ‘enum’ keyword in java**

**main() method in java**

* main() method Introduction
* Why main() method is 'public'
* Why main() method is 'static'
* Why main() method has 'void' Return Type
* Why main() method required 'parameter'
* Why main() method required 'String[]' parameter
* List of valid and Invalid Syntaxes of main() method
* More than one main() in Single Java Application
* main() method Overloading and Overriding
* main() method in Inheritance

**Association in java**

* Introduction To Relationships
* One-To-One Association through Constructor Dependency Injection
* One-To-One Association through setter method Dependency Injection
* One-To-Many Association through Constructor Dependency Injection
* One-To-Many Association through setter Dependency Injection
* One-To-Many Association Internal Data Representation
* Many-To-One Association through Constructor Dependency Injection
* Many-To-One Association through setter Dependency Injection
* Many-To-Many Association through Constructor Dependency Injection
* Many-To-Many Association through Setter Method Dependency Injection
* Many-To-Many Association through Setter Method Dependency Injection
* Composition Vs Aggregation

**Inheritance in java**

* Inheritance Introduction
* Types of Inheritance
* Static Context in Inheritance
* Instance Context in Inheritance
* Static Context and Instance Context Mix in Inheritance

**Super keyword in java**

**Class level type casting in java**

* Up Casting Part
* Down Casting Part

**USES a relationship in java**

**Polymorphism**

* Method Overloading
* Method Overriding
* Rules and Regulations for Method Overriding

**Abstract class in java**

* Abstract Methods and Abstract classes Introduction
* Concrete Method Vs Abstract Method
* Concreate class Vs Abstract Class
* Abstract Class

**Interfaces in java**

* Interfaces Part
* Syntaxes between classes, abstract classes and Interfaces

**Marker interfaces and adapt classes**

* Marker Interfaces : Serializable
* Marker Interfaces : Cloneable
* Adapter Classes

**Object cloning in java**

* Object Cloning Introduction
* Shallow Cloning
* Deep Cloning

**Instanceof operator in java**

**Java Back-End Server Application Development (16 Hours)**

**JDBC in Java EE Environments**

* Overview of the JDBC API
* The Data Access Object pattern
* Using CDI to inject a JDBC resource in a Java EE component

**Java Persistence API**

* Persistence contexts and persistence units
* Create, read, update and delete operations with JPA
* Entities and the entity manager
* Object-relational mapping
* Create typed queries in JPA with JPQL

**HTTP / REST Webservices / Postman tool / API Security (20 hours)**

**API introduction**

* Introduction to web application architecture
* Introduction to APIs
* Introduction to Web-Services
* How does an API works
* What is API testing?
  + What does API Testing involve
  + Validation techniques used in API Testing
  + API Testing Steps
  + Understanding URI, End points, Resources, Http verbs
  + GUI tools available for API Testing
  + Command-line tools available for API Testing
  + Best Practices for API Testing
* Advantages of API
* API vs Web-Services
* Introduction to API architecture, REST API, SOAP API
* Understanding how REST API architecture works
* Understanding how SOAP API architecture works
* Understanding the HTTP methods GET, POST, PUT, DELETE, PATCH, OPTIONS, HEAD, and Few more

**Introduction to Postman API Testing Tool**

* What is Postman tool
* Installation of Native Postman tool
* Installation of Postman tool as Chrome Add-on
* Introduction to Postman landscape
* Introduction to Postman Settings

**API Testing using Postman**

* Creating First API request using Postman
* Functional Testing of Web Services
* Understanding History
* Setting up Test Project with REST APIs
* Validating responses with Postman client
* Validating the body of the first JSON response
* Examples of validating response headers and status codes
* Understanding query parameters on sending request Validating status code

**Maven (4 Hours)**

**Introduction to Maven**

* Overview of Maven
* Pom.xml
* Build life cycle
* Phases and goals

**Spring Boot Framework (16 Hours)**

**Introduction to Spring**

* Spring Modules

**Spring Core**

* Introduction to IOC
* Types of DI
* Setter VS Constructor
* Collection DI
* Bean Inheritance
* Inner Beans
* Bean Scopes
* Inner Beans
* Bean auto wiring
* Static Factory Method
* Instance Factory Method
* Bean Lifecycle

**Spring AOP**

* AOP Concepts
* Programmatic VS Declarative AOP
* Programmatic AOP
* Types of Advices
* Types of Pointcuts

**Spring MVC**

* Introduction to Spring MVC
* Handler Mapping
* Controllers
* Validations
* Views
* Form tags
* **Introduction to Spring Boot**
* Intro to Spring Boot - What is Spring Boot and What It Does
* Spring Boot Hello World / Spring Application
* Download and Install STS IDE
* Brief maven Overview
* Spring Beans & Dependency Injection
* Configuration
* Configuration Properties
* Application Properties & YAML Configuration

**Spring Boot Annotations**

* SpringBootApplication
* EnableAutoConfiguration
* SpringBootConfiguration
* Embedded server and its uses
* What is Spring Boot Actuator

**RESTFUL WEB SERVICES**

* REST Overview (Characteristics/Capabilities, URI Templates, REST vs SOAP
* REST and Spring MVC
* Spring support for REST
* @RequestMapping/@PathVariable, @RequestBody, @ResponseBody, HTTP Method conversion
* URI Templates and @PathVariable
* Writing RESTful Controllers / @RestController

**Accessing Data with Spring Boot and Database Support**

* Data Access Introduction
* Spring Data JDBC
* Basic Auto-configuration - Data source
* Configuration Properties
* Spring Boot's JPA Support - spring-boot-starter-data-JPA

**SPRING BOOT SECURITY**

* Adding the spring boot security starter
* Apply the security configuration by writing our own authorization and authentication

**AWS Fundamental (40 Hours)**

**Introduction to Amazon Web Services**

* Introduction to AWS Cloud
* Security in the AWS Cloud
* Hosting the employee directory application in AWS
* Hands-On Lab: Introduction to AWS Identity and Access Management (IAM)

**AWS Compute**

* Compute as a service in AWS
* Introduction to Amazon Elastic Compute Cloud
* Amazon EC2 instance lifecycle
* AWS container services
* What is serverless?
* Introduction to AWS Lambda
* Choose the right compute service

**AWS Networking**

* Networking in AWS
* Introduction to Amazon Virtual Private Cloud (Amazon VPC)
* Amazon VPC routing
* Amazon VPC security

**AWS Storage**

* AWS storage types
* Amazon EC2 instance storage and Amazon Elastic Block Store (Amazon EBS)
* Object storage with Amazon S3
* Choose the right storage service

**Databases**

* Explore databases in AWS
* Amazon Relational Database Service
* Purpose-built databases
* Introduction to Amazon DynamoDB
* Choose the right AWS database service
* Hands-On Lab: Implement and manage Amazon DynamoDB

**Monitoring, Optimization, and Serverless**

* Monitoring
* Optimization
* Alternate serverless employee directory application architecture
* Hands-On Lab: Configure High Availability for Your Application

**Building a Web Application on AWS**

* Discuss the architecture of the application you are going to build during this course
* Explore the AWS services needed to build your web application
* Discover how to store, manage, and host your web application
* Module 3: Getting Started with Development on AWS
* Describe how to access AWS services programmatically
* List some programmatic patterns and how they provide efficiencies within AWS SDKs and
* AWS CLI
* Explain the value of AWS Cloud9

**Getting Started with Permissions**

* Review AWS Identity and Access Management (IAM) features and components permissions
* to support a development environment
* Demonstrate how to test AWS IAM permissions
* Configure your IDEs and SDKs to support a development environment
* Demonstrate accessing AWS services using SDKs and AWS Cloud9

**Getting Started with Storage**

* Describe the basic concepts of Amazon S3
* List the options for securing data using Amazon S3
* Define SDK dependencies for your code
* Explain how to connect to the Amazon S3 service
* Describe request and response objects

**Processing Your Storage Operations**

* Perform key bucket and object operations
* Explain how to handle multiple and large objects
* Create and configure an Amazon S3 bucket to host a static website
* Grant temporary access to your objects
* Demonstrate performing Amazon S3 operations using SDKs

**Getting Started with Databases**

* Describe the key components of DynamoDB
* Explain how to connect to DynamoDB
* Describe how to build a request object
* Explain how to read a response object
* List the most common troubleshooting exceptions

**Processing Your Database Operations**

* Develop programs to interact with DynamoDB using AWS SDKs
* Perform CRUD operations to access tables, indexes, and data
* Describe developer best practices when accessing DynamoDB
* Review caching options for DynamoDB to improve performance
* Perform DynamoDB operations using SDK

**Processing Your Application Logic**

* Develop a Lambda function using SDKs
* Configure triggers and permissions for Lambda functions
* Test, deploy, and monitor Lambda functions

**Managing the APIs**

* Describe the key components of API Gateway
* Develop API Gateway resources to integrate with AWS services
* Configure API request and response calls for your application endpoints
* Test API resources and deploy your application API endpoint
* Demonstrate creating API Gateway resources to interact with your application APIs

**Building a Modern Application**

* Describe the challenges with traditional architectures
* Describe the microservice architecture and benefits
* Explain various approaches for designing microservice applications
* Explain steps involved in decoupling monolithic applications
* Demonstrate the orchestration of Lambda Functions using AWS Step Functions

**Granting Access to Your Application Users**

* Analyze the evolution of security protocols
* Explore the authentication process using Amazon Cognito
* Manage user access and authorize serverless APIs
* Observe best practices for implementing Amazon Cognito
* Demonstrate the integration of Amazon Cognito and review JWT tokens

**Deploying Your Application**

* Identify risks associated with traditional software development practices
* Understand DevOps methodology
* Configure an AWS SAM template to deploy a serverless application
* Describe various application deployment strategies
* Demonstrate deploying a serverless application using AWS SAM

**Observing Your Application**

* Differentiate between monitoring and observability
* Evaluate why observability is necessary in modern development and key components
* Understand CloudWatch’s part in configuring the observability
* Demonstrate using CloudWatch Application Insights to monitor applications
* Demonstrate using X-Ray to debug your applications