

CORE CONCEPTS

Introduction to Java

- L Evolution of Java
- L Object Oriented Programming Structure
- L Client side Programming & its requirements
- L Platform Independency & Portability
- L Security
- L Relation b/w JVM, JRE and JDK
- L Description of a Simple Java Program
- L Introduction to JAR format
- L Naming Conventions
- L Data types & Type casting

OOPS Implementation

Classes and Objects

- L Defining attributes and methods
- L Implementing data encapsulation
- L Relation b/w objects and reference variables
- L Constructors and Anonymous block
- L Method Overloading
- L Static Data members, Block & methods

Understanding Memory Structure

- L Stack
- L Heap
- L Class & Method area

Understanding Class loading & Execution flow

- L Static vs Dynamic Class loading
- L Implicit vs Explicit class loading
- L Sequence of operations performed at the time of class loading

Argument Passing Mechanism

- L Passing primitive arguments
- L Passing objects
- L Wrapper Classes & their use

Usage of this keyword

- L Referencing instance members
- L Intra class constructor chaining
- L Method chaining

Inheritance & code reusability

- L Extending classes for code reusability
- L Usage of super keyword
- L Method Overriding
- L Object class and its roles

Inheritance & Runtime Polymorphism

- L Static & Dynamic binding
- L Inheritance and Is-A relation
- L Runtime Polymorphism and Generalization
- L Abstract classes & methods
- L Final Keyword

Interfaces and Role based Inheritance

- L Difference b/w Feature & Role based Inheritance
- L Difference b/w Static & Dynamic classing Environment
- L Difference b/w classes & interfaces
- L Use fo interfaces in real scenarios

Implementing Has-A relation

- L Difference b/w Aggregation & Composition
- L Nested classes and their role
- L Inner classes
- L Anonymous Inner classes

Package & Scopes

- L Need of Packages
- L Associating classes to Packages
- L Understanding Classpath environment variable
- L Import Keyword and Feature of static import
- L Public, protected, private & default scope
- L Private Inheritance and its use
- L Examples of private inheritance

Exception Handling

- L Difference b/w exception and error
- L Exception Handling & Robustness
- L Common Exceptions and Errors
- L Try and catch block
- L Exception handlers & importance of their order
- L Throw keyword and its usages
- L Checked and Unchecked Exceptions
- L Role of finally
- L Creating User defined Exceptions

Collection Framework

- L Role and Importance of Collection Framework

- [L Use of List & Set based collection](#)
- [L Use of Iterator & ListIterator](#)
- [L Use of Maps](#)
- [L Searching elements in List, Hash and Tree based collections](#)
- [L Role of equals and hashCode\(\) methods](#)
- [L Role of Comparable and Comparator Interfaces](#)
- [L Thread Safety and Vector](#)
- [L Difference b/w Enumeration and Iterator](#)
- [L Type safety and Generics](#)
- [L Common algorithms and Collections class](#)
- [L Using Properties class for managing properties files](#)

Database Connectivity Using JDBC 4.x

- [L Overview of native and ODBC Drives](#)
- [L Introduction to JDBC](#)
- [L Type of JDBC drivers](#)
- [L Using different type of drivers](#)
- [L Defining properties based Connection Factory](#)

Performing basic database operations

- [L Insert](#)
- [L Delete](#)
- [L Update](#)
- [L Select](#)

Prepared Statement

- [L Difference b/w Statement & Prepared Statement](#)
- [L Setting Query parameters](#)
- [L Executing Queries](#)

Callable Statement

- [L Creating PL/SQL Stored procedures and functions](#)
- [L Creating Callable statements](#)
- [L Executing procedures & functions](#)

Misc

- [L Batch Updation](#)
- [L Transacting Queries](#)
- [L Programmatic initialization of database](#)
- [L ResultSetMetaData](#)
- [L DatabaseMetaData](#)
- [L Introduction to SpringBoot](#)
- [L IOC & DI](#)
- [L Web MVC](#)

- [Spring Data](#)
- [Creating REST API using SpringBoot](#)

Introduction

- [Course Objectives](#)
- [Course Outline](#)
- [What is Angular](#)
- [Why use Angular](#)

Introduction to TypeScript

- [Why Use TypeScript](#)
- [Basic Types](#)
- [Classes and Interfaces](#)
- [Type Definitions](#)
- [Compiling TypeScript](#)

Environment Setup

[Node / NPM](#)

- [TypeScript](#)
- [Application File Structure](#)
- [Angular CLI](#)
- [Code Editors](#)

Getting Started

- [Our First Component](#)

Modules

- [Why use Modules](#)
- [NgModule](#)
- [Declarations](#)
- [Providers](#)
- [Imports](#)
- [Bootstrapping](#)
- [The Core Module](#)
- [Shared Modules](#)

Components

- [Introduction to Components](#)
- [Component Architecture Patterns](#)
- [Decorator Metadata](#)
- [State & Behaviour](#)
- [Inputs and Outputs](#)

Templates

- [Inline vs External](#)
- [Template Expressions](#)

- [Data Bindings](#)
- [*ngIf else and *ngFor](#)
- [Built-in Structural Directives](#)
- [Built-in Attribute Directives](#)

Custom Directives

- [Types of Directive](#)
- [Create your own Structural Directive](#)
- [Create your own Attribute Directive](#)

Pipes

- [Built-in Pipes](#)
- [Custom Pipes](#)

Services

- [Introduction to Services](#)
- [Building a Service](#)

Dependency Injection

- [Introduction to Dependency Injection](#)
- [Injectors & Providers](#)
- [Registering Providers](#)

Lifecycle Hooks

- [Component LifeCycle](#)
- [Using ngOnInit](#)
- [All lifecycle Hooks](#)

Routing

- [The Component Router](#)
- [Defining Routes](#)
- [Navigation](#)
- [Route Params](#)
- [Child Routes](#)
- [Route Guards](#)

Template-driven Forms

- [Introduction to forms](#)
- [Template-driven forms](#)
- [Validation](#)

Asynchronous Operations

- [Introduction to Async](#)
- [Promises](#)
- [Observables](#)
- [HTTP Request / Response](#)

Testing

- [Introduction to Testing](#)
- [Unit Testing](#)

- E2E Testing

Third –Party NPM Package

- Social Login Authentication
- Cookies and Local Storage
- Angular File Upload
- Angular Pagination
- Angular Shopping Cart

Building & Deployment

- Environments
- Builds

- Deployment on a Server
- Covering All The Concepts