**Core Java**

Fundamentals

* Java & JVM and It's futures
* Programming fundamental (Data Types, Arrays, Operators, Arrays and Flow Control statements if, else, switch, while, do while, for and for each.)

Basics

* Variables
* Methods
* Class
* Object
* Coding standards
* Main methods
* packages
* imports
* Access specifiers (Public, Private, protected and default)

Object Oriented Concepts

* Inheritance
* Abstraction
* Encapsulation
* Polymorphism (Method Over loading and Method Overriding )
* Constructors

Implementations of OOPs

* Abstract class
* Interfaces
* var-args
* Inner classes
* super and this key words
* final key word

Predefined Classes

* Object class
* Garbage Collector
* String
* String buffer and String builder
* enum
* Wrapper classes,
* Auto boxing, Auto unboxing
* Object Type casting

Exception Handling

* Exception overview
* Default Exception Handler
* Handling Exception using try and catch
* finally
* throws
* throw
* Custom Exceptions

Multi-Threading & Concurrency API

* Threads overview
* Thread creation
* Thread Life cycle
* Threads class methods (priority, yeild, join)
* Synchronization
* Intra thread communication (Wait, notify methods)

Collection Framework

* Collection framework interfaces List and set overview
* List implemented classes (ArrayList,Vector...)
* Set implemented Classes (HashSet,TreeSet...)
* Iterator , ListIterator and for each
* Comparator and Comparable interface
* Map implemented classes (HashMap,Hashtable. )
* Collections class
* Generics

File Operations

* Create File , Directory..
* Read and Write file data
* I/O Streams
* Serialization

Java 8,9,10 Features like Lambda expression, functional interface, etc….

**REST API**

* REST API Training Course
* Vocabulary: The Primer's Primer
* REST
* What REST Is Not
* REST Services
* Mapping CRUD To HTTP
* Conventions API URL structure
* RESTFUL HTTP Methods
* Passing Parameters
* HTTP Response Codes
* Return Values
* JSON - a universal format for the exchange of information
* JSON - what does it look like?
* JSONP & CORS
* Passing Parameters (part 2)
* More Simple APIs
* Examples of popular APIs
* Creating a REST API
* Business Object - mapping the class
* Controller methods
* Representation of a business object as JSON
* HTTP Headers - as metadata
* Testing the API using Postman
* Authentication mechanisms

**Spring Boot, Spring JPA & Micro Services**

* Introduction to spring boot
* Building Spring Boot Application
* Rest Annotation with In Memory Database & CRUD Operations
* Rest Annotation with Relation DB
* Spring JPA Repository Concepts
* Actuator Concepts
* Spring Boot Custom Logging
* Spring Boot Profile Components
* Auto Configuration
* Integration with Spring Web
* Spring Boot Security
* Database Concepts
* Core Concepts
* Micro Services
* Micro Services Design Considerations
* Fault Tolerance Concepts
* API Gateway
* Messaging Queue Concepts
* Oatuh2 Concepts

**React JS**

* Introduction to React
  1. What is React?
  2. Why React?
  3. React version history
  4. React 16 vs React 15
  5. Just React – Hello World
  6. Using create-react-app
  7. Anatomy of react project
  8. Running the app
  9. Debugging first react app
* Templating using JSX

1. Working with React. createElement
2. Expressions
3. Using logical operators
4. Specifying attributes
5. Specifying children
6. Fragments

* About Components

1. Significance of component architecture
2. Types of components
3. Functional
4. Class based
5. Pure
6. Component Composition

* Working with state and props

1. What is state and it significance
2. Read state and set state
3. Passing data to component using props
4. Validating props using propTypes
5. Supplying default values to props using defaultProps

* Rendering lists

1. Using react key prop
2. Using map function to iterate on arrays to generate elements

* Event handling in React

1. Understanding React event system
2. Understanding Synthetic event
3. Passing arguments to event handlers

* Understanding component lifecycle and handling errors

1. Understand the lifecycle methods
2. Handle errors using error boundaries

* Working with forms</str

1. Controlled components
2. Uncontrolled components
3. Understand the significance to default Value prop
4. Using react ref prop to get access to DOM element

* Context

1. What is context
2. When to use context
3. Create Context
4. Context.Provider
5. Context.Consumer
6. Reading context in class

* Code-Splitting

1. What is code splitting
2. Why do you need code splitting
3. React.lazy
4. Suspense
5. Route-based code splitting

* Hooks

1. What are hooks
2. Why do you need hooks
3. Different types of hooks
4. Using state and effect hooks
5. Rules of hooks

* Routing with react router

1. Setting up react router
2. Understand routing in single page applications
3. Working with BrowserRouter and HashRouter components
4. Configuring route with Route component
5. Using Switch component to define routing rules
6. Making routes dynamic using route params
7. Working with nested routes
8. Navigating to pages using Link and NavLink component
9. Redirect routes using Redirect Component
10. Using Prompt component to get consent of user for navigation
11. Path less Route to handle failed matches

* Just Redux

1. What is redux
2. Why redux
3. Redux principles
4. Install and setup redux
5. Creating actions, reducer and store

* Immutable.js

1. What is Immutable.js?
2. Immutable collections
3. Lists
4. Maps
5. Sets

* React Redux

1. What is React Redux
2. Why React Redux
3. Install and setup
4. Presentational vs Container components
5. Understand high order component
6. Understanding mapStateToProps and mapDispatchtToProps usage

**Points to Note**

* **I will come on time**
* **In every class write the example and show it practically with well explanation.**
* **We are explaining and develop the each topic with taking the some real time project scenarios. (Just like develop a mini project with those topics)**
* **Total Course – 75 to 90 classes**
* **I will share all softwares.**
* **Each class has taken minimum of 1 hours to 1:30 hours.**
* **I will send the examples to the Students after completion of each class and ask them to practice the same.**
* **I will give the Assignments and Interview Questions to the students after completion of each topic.**
* **I will monitor the each Students assignment.**