**Full Stack Training**

* **Core java With JDBC with Java 8 Features --> 8 days**
* **HTML5/CSS3,Bootstrap and JavaScript --> 5 days**
* **ES6, Angular/React JS --> 5 days**
* **SQL Oracle /PL SQL --> 5 Days**
* **Testing/ Selenium --> 3 Days**
* **Devops - Unix, Git, Cloud, AWS (Overview), Jenkin, --> 6 Days**
* **Python ---> 3 Days**
* **Project Assessments --> 6 Days**

**Total Number of Days -- 45 Days**

**Program Duration:** 8 days

**Day 1**

* Introduction to Object-Oriented technology
  + Object Oriented concepts
  + What is Object-Oriented Programming?
  + Why Object-Oriented Programming?
* Objects and Classes
  + What is an Object?
  + What is a Class?
* Principles in Object-Oriented technology
  + Object-Oriented Principles like Abstraction, Encapsulation, Modularity, Hierarchy and Polymorphism
* Some more concepts in OOP
  + Static Members
  + Abstract Class
  + Interface
  + Packages
* Introduction to Java
  + Introduction to Java
  + Features of Java
  + Evolution in Java
  + Developing software in Java
* Eclipse 4.4 (Luna) as an IDE
  + Installation and Setting up Eclipse
  + Introduction to Eclipse IDE
  + Creating and Managing Java Projects
  + Use of Java docs
  + Miscellaneous  Options
* Language Fundamentals
  + Keywords
  + Primitive Data Types
  + Operators and Assignments
  + Variables and Literals
  + Flow Control: Java’s Control Statements
  + Best Practices
* Classes and Objects
  + Classes and Objects
  + Packages
  + Access Specifiers
  + Constructors - Default and Parameterized
  + this reference
  + using static keyword
  + Best Practices
* Exploring Java Basics
  + The Object Class
  + Wrapper Classes
  + Type casting
  + Using Scanner Class
  + String Handling
  + Date and Time API
  + Best Practices

**Day 2**

* Inheritance and Polymorphism
  + Inheritance
  + Using super keyword
  + InstanceOf Operator
  + Method & Constructor overloading
  + Method overriding
  + @override annotation
  + Using final keyword
  + Best Practices
* Abstract Classes and Interfaces
  + Abstract class
  + Interfaces
  + default methods
  + static methods on Interface
  + Anonymous classes
  + Runtime Polymorphism
  + Best Practices
* Exception Handling
  + Introduction
  + Exception Types
  + Exception Hierarchy
  + Try-catch-finally
  + Try-with-resources
  + Multi catch blocks
  + Throwing exceptions using throw
  + Declaring exceptions using throws
  + User defined Exceptions
  + Best Practices

**Day 3**

* Array
* One dimensional array
* Multidimensional array
* Using varargs
* Using Arrays class
* Best Practices
* Collection
  + Collections Framework
  + Collection Interfaces
  + Implementing Classes
  + Iterating Collections (using foreach & iterator)
  + Comparable and Comparator
  + Hashtable, HashMap, TreeMap
  + Best Practices
* Generics
  + Generics
  + Writing Generic Classes
  + Using Generics with Collections
  + Best Practices

**Day 4**

* Muliti Threading
  + Understanding threads
  + Thread life cycle and Scheduling threads- Priorities , Sleep(),join()
  + Consumer Producer problem
  + Inter Thread communication : wait, notify, notifyAll methods
  + Synchronization concept
* File IO
  + Overview of I/O Streams
  + Types of Streams
  + The Byte-stream  I/O hierarchy
  + Character Stream Hierarchy
  + Buffered Stream
  + The File class
  + The Path class
  + Object Stream
  + Best Practices
* Property Files
  + What are Property Files?
  + Types of Property files
  + User defined Properties
* Introduction to Junit 4
  + Why testing?
  + Why use JUnit?
  + Installing and Running JUnit
  + Understanding JUnit Framework
  + Testing with Junit i.e Test Driven Development
  + Advanced Testing Concepts
  + Advanced Testing concepts
  + Test Suites
  + Parameterized Tests
  + Mocking Concepts using EasyMock Framework

**Day 8**

* Lambda Expressions
  + Introduction
  + Writing Lambda Expressions
  + Functional Interfaces
  + Types of Functional Interfaces
  + Method reference
* Stream API
  + Introduction
  + Stream API with Collections
  + Stream Operations

**HTML 5, CSS 3 with Bootstrap & Javascript**

**Program Duration:** 5 days

**Contents:**

**HTML 5:**

* HTML Basics
  + Understand the structure of an HTML page.
  + New Semantic Elements in HTML 5
  + Learn to apply physical/logical character effects.
  + Learn to manage document spacing.
* Tables
  + Understand the structure of an HTML table.
  + Learn to control table format like cell spanning, cell spacing, border
* List
  + Numbered List
  + Bulleted List
* Working with Links
  + Understand the working of hyperlinks in web pages.
  + Learn to create hyperlinks in web pages.
  + Add hyperlinks to list items and table contents.
* Image Handling
  + Understand the role of images in web pages
  + Learn to add images to web pages
  + Learn to use images as hyperlinks
* Frames
  + Understand the need for frames in web pages.
  + Learn to create and work with frames.
* HTML Forms for User Input
  + Understand the role of forms in web pages
  + Understand various HTML elements used in forms.
  + Single line text field
  + Text area
  + Check box
  + Radio buttons
  + Password fields
  + Pull-down menus
  + File selector dialog box
* New Form Elements
  + Understand the new HTML form elements such as date, number, range, email, search and datalist
  + Understand audio, video, article tags

**CSS 3**

* Introduction to Cascading Style Sheets 3.0
  + What CSS can do
  + CSS Syntax
  + Types of CSS
* Working with Text and Fonts
  + Text Formatting
  + Text Effects
  + Fonts
* CSS Selectors
  + Type Selector
  + Universal Selector
  + ID Selector
  + Class selector
* Colors and Borders
  + Background
  + Multiple Background
  + Colors RGB and RGBA
  + HSL and HSLA
  + Borders
  + Rounded Corners
  + Applying Shadows in border

**Javascript**

* JavaScript Language
  + Data Types and Variables
  + JavaScript Operators
  + Control Structures and Loops
  + JavaScript Functions
* Working with Predefined Core Objects
  + Data Types in JavaScript
  + String Objects
  + URL String Encoding and Decoding
  + Math Properties
  + Math Objects
  + Date Objects
  + Date and Time Arithmetic
* Working with arrays
  + Arrays object, its properties and methods
* Document Object Model
  + Understand the JavaScript Object Model
  + Understand the Window object
* Working With Document Object
  + Document Object and its properties, methods and events
* Working with Form Object
  + Form Object Properties, Methods & Event Handlers
  + Text-Related Objects
  + Button Objects
  + Check Box and Radio Objects
  + Select Objects

Validate Data and Form Submission

**Angular 7**

**Program Duration:** 5 days

**Contents:**

* Introduction to Angular 7
* What is Angular 7?
* Why Angular 7?
* What is nodeJS?
* Scope and Goal of Angular 7
* Installing and using Angular 7
* Building Blocks of an Angular 7  Application
* Difference between Angular 1.x Angular 2.x, 4.x, 5.x, 6.x and 7.x
* A Basic Angular 4 Application
* Working with Angular 5 with VSC
* Components
* What is a component?
* Developing a simple component.
* Templates for a component.
* Component lifecycle
* Data Binding
* What is data binding
* One way data binding
* Two way data binding
* Nested component
* Event binding
* Directives
* What are directives?
* Types of directives - component, structural and attribute
* Creating a basic directive
* Handling event & Binding input in attribute directive
* Creating your own structural directive
* Using the structural directive
* Binding input to a structural directive
* Working with Forms
* Forms in Angular 5
* Template Driven Form
  + Create the component that controls the form
  + Create a template with the initial form layout
  + Bind data properties to each form input control with the ngModel two-way data binding syntax
  + Add the name attribute to each form input control
  + Add custom CSS to provide visual feedback
  + Show and hide validation error messages
  + Handle form submission with ngSubmit
  + Disable the form’s submit button until the form is valid
  + Resetting the form
* Model Driven Forms
* Reactive Forms Introduction
* More Form Controls
* Form Control Properties
* setValue and patchValue
* Validating Form Elements
* A Basic Angular Form
* Binding Input Fields
* Displaying Form Validation State & Field Validation State
* Displaying Validation State Using Classes
* Disabling Submit when Form is Invalid
* Service and Dependency Injection
* What is a service?
* Injecting a service to a component
* Application wide dependency injection
* @Injectable classes
* Multiple service instances
* @Optional and @Host
* HTTP Client
* The new HTTP providers
* Injecting the providers
* GET call
* Handling error
* About Observables
* POST request
* Working with headers
* Sequential calls & parallel calls
* Pipe
* What is a pipe?
* Passing parameters to a pipe
* Chaining pipes
* Developing a custom pipe
* Routing
* Why use routing?
* Defining a route table
* New Routing architecture Angular 7
* Navigation using hyperlink & code
* Supplying parameters to a route URL
* Animation using Angular 7
* Creating Angular5-CRUD Application with Web

Services

**React JS**

**Program Duration:** 3 days

**Contents**:

* **Introduction of React JS**
  + Javascript MVC
  + View Technology
  + Introduction to ReactJS
  + React and SPA
  + React Introduction
  + Why React?
  + Basic Setup
  + React Advantages
  + React Official Website
  + React CDN Hosting
  + First Example – Hello World
* **JSX and the Virtual DOM**
  + **What is a Virtual DOM?**
  + **ReactElement**
  + **Rendering Our ReactElement**
  + **Rendering Our ReactElement**
  + **Adding Text (with children)**
  + **ReactDOM.render()**
  + **JSX**
  + **JSX Creates Elements**
  + **JSX Attribute Expressions**
  + **JSX Conditional Child Expressions**
  + **JSX Boolean Attributes**
* **Components ,Data & Servers**
  + **The Server API**
  + **JSON endpoints**
  + **Loading state from the server**
  + React Top-Level API
  + React.createElement
  + ReactDOM.render
  + ReactDOMServer.renderToString
  + ReactDOMServer.renderToStaticMarkup
  + React.createClass
  + React.Children.count
  + React.Children.map
  + React.Children.forEach
  + **ReactComponent**
  + **render() Returns a ReactElement Tree**
  + **Getting Data into render()**
  + **props are the parameters**
  + **PropTypes**
  + **Default props with getDefaultProps()**
  + **Context**
* **Lists and Forms & Events**
  + **Introduction of Forms**
  + **The Basic**
  + React Element
  + React Component
  + Working with props(Properties)
  + Prop Validation
  + Static Methods in React
  + Nested Components
  + React and CSS
  + Adding key for Dynamic Children
  + **Events and Event Handlers**
  + **Text Input**
  + **Accessing User Input With refs**
  + **Using User Input**
* **State, Life Cycle, and Events**
* Unidirectional data flow
* Component Types
* props or state?
* React Architecture
* React component's Life Cycle
* React component - Life cycle phases
* React component - Life cycle methods execution sequence
* Mixins
* **Routing**
* What's in a URL?
* React Router's core components
* React Router v4
* Building the components of react-router
* Middleware
* Middleware Applicability
* React Router Configuration
* React Router with Redux
* Navigate with React Router
* **Building React Apps with Flux/Redux** 
  + **Node Package Manager**
  + **EventEmitter**
  + **Creating an EventEmitter**
  + **Gulp-JavaScript Task Runner**
  + **Gulp API**
  + **Creating React Component modules**
  + **Creating gulpfile.js**
  + **Flux Introduction**
  + **Flux - Action**
  + **Flux - Dispatcher**
  + **Flux - Store**
  + **React View (Controller View)**
  + **Flux flow**

**SQL and PL-SQL Syllabus**

**Day 1**

Overview of Database Management and Architecture

* What is Database system
* Why Database

Introduction Relational Database

* Database Models
* Relational Model
* Relational Model paradigm
* Keys
* Domain
* Relation Operators
* Integrity
* Attribute Constraints
* Database Constraints

Normalization

* Introduction to Normalization
* Functional Dependency and Non-loss Decomposition
* First Normal Form
* Second Normal Form
* Third Normal Form
* De-normalization

Retrieving Data using SQL(Select Clause)

* Tables
* The Select statement
* Selecting all columns
* Selecting specific column
* Arithmetic Operators and Expression
* Null values
* Using Column Aliases
* Concatenation Operator
* Eliminating duplicating rows.
* Using the Where clause
* Comparison Operator
* Logical Operator
* Precedence of Logical Operator
* Between operator
* The In Operator
* Wild card matching

**Day 2**

Function in Oracle

* Single row function
* Character function
* Date function
* Case and Decode function

DML (Data Manipulation Language

* Insert
* Delete
* Update
* Merge

Joining Tables

* Need for join
* Types of join
* Ambiguous Column
* Outer join
* Inner join
* Self join

DDL (Data Definition Language)

* Creating tables
* Alter the table
* Rename the table
* drop the table
* Add/Drop/Modify the column

**Day 3**

Constraints

* Column Level Constraint
* Table Level Constraint
* Not Null Constraints
* Unique Constraints
* Primary key Constraints
* Check Constraints
* Foreign Key Constraints
* Alter Table With Constraints
* Enable Constraints
* Disable Constraints
* Cascade Constraints

Group Functions:

* Using NVL with group function
* Having with group function

Sub Queries:

* Single Row Sub Query
* Multiple Row Sub Query
* Group function using Sub Query
* All with Sub Query
* Not in with Sub Query

Creating VIEW

* Simple view
* Where using View
* DML operation using View
* Force using View
* Column Alias using View
* Group function using View

Creating Sequence, Index, Synonym

* NEXTVAL and CURRVAL Pseudocolumn
* How to use Sequence
* Index
* Unique Index

Using SET Operator

* Union
* Union All
* Intersect
* Minus

**Day 4**

Basics of PL/SQL:

* Overview and benefits of PL/SQL
* PL/SQL architecture
* PL/SQL and SQL\*Plus
* Types of PL/SQL blocks

PL/SQL structures:

* Simple Anonymous block
* Identifiers
* Types of identifiers
* Control structures
* PL/SQL records
* Recognizing the Basic PL/SQL Block and Its Sections
* Describing the Significance of Variables in PL/SQL
* Distinguishing Between PL/SQL and Non-PL/SQL Variables
* Declaring Variables and Constants
* Executing a PL/SQL Block
* The %TYPE Attribute
* Bind variables
* Sequence in PL/SQL Expression
* Executable Statements
* PL/SQL Block syntax
* Comment the code

Boolean logic in PL/SQL:

* Conditional processing using if statement
* Case Statement and Case Statements
* Constructing and Identifying Different Loop Statements
* Controlling Block Flow Using Nested Loops and Labels
* Using Logic Tables
* If-then-else structure
* Continue and Break statement

Composite Data Types

* PL/SQL Records
* The %ROWTYPE Attribute
* Insert and Update with PL/SQL Records.
* Index by Tables.
* Index by Table Methods,
* Use Index by Table of Records

Cursors in PL/SQL:

* Cursor basics
* Declare the cursor
* Open the cursor
* Fetch data from the Cursor
* Close the Cursor
* Cursor for loop
* The %NOTFOUND and %ROWCOUNT Attribute
* For Update Clause and Where Current Clause

**Day 5**

PL/SQL tables:

* Defining PL/SQL tables
* Reasons to use PL/SQL tables
* Populating a PL/SQL table
* Retrieving from a PL/SQL table

Exception Handling:

* Handle Exception with PL/SQL
* Trap pre-defined and non predefined Oracle Server Errors.
* User- Defined Exception
* Propagate Exception
* RAISE\_APPLICATION\_ERROR procedure

Stored procedures and functions

* Create a modularized and layered subprogram Design
* The PL/SQL execution Environment
* Difference between Anonymous blocks and subprograms
* Create, Call and Remove stored procedure
* Implement Procedures parameters and parameter modes (IN,OUT,IN-OUT)
* View Procedure Information
* Stored functions and debugging subprogram
* Advantage of using Stored functions
* Step to create a stored functions
* Invoke User-defined function in SQL statements
* Restriction when calling functions
* Difference between Stored procedure and Stored function

Packages

* Advantage of packages
* Component of package
* Develop a package
* Enable visibility of package's component
* Create the package specification and body using the SQL Create Statement
* Invoke the Package construct
* View the PL/SQL source code using the Data Dictionary
* Deploying packages
* Overloading subprograms in PL/SQL
* Implement package function in SQL and Restrictions
* Invoke PL/SQL Tables of Records in Packages

**Testing and Selenium**

* What is software
* Software technologies
* What is web application
* Web Application technologies

Software Testing Life Cycle (STLC)

* Understand Requirement
* Create the test cases
* Manual Testing
* Automation Testing
* Test report
* Summary

Manual Testing

* Writing test scenarios
* Test planning
* Test case design
* Test data identification
* Reviewing and Execution of Test Cases/script

Automation Testing

* What is automation testing
* Use of automation testing
* Explanation of selenium and its advantage
* Different between Selenium of QTP

Component of Selenium

* Selenium IDE
* Selenium RC
* Selenium Grid
* Selenium Grid
* Selenium Webdriver / version 2.0

Locator of Selenium

* Tools of identity elements and objects.
* Firebugs
* IE developer tools
* Google chrome developer tools
* Locating element by Id, name, link text , XPath

Introduction of Selenium IDE

* Main features of Selenium IDE
* Selenium IDE Icons
* Installing Selenium IDE
* First test with Selenium IDE

**Selenium IDE concepts**

* Building Test cases using Selenium IDE
* Creating test suites
* Adding Selenium IDE commands
* Selenium IDE Menus
* Selenium IDE commands (asserting, verifying , wait, and storing the elements)
* IDE context menu

**Selenium IDE Commands**

* goBack
* refresh
* keys simulation
* mouseOver
* Highlight

**Selenium Webdriver**

* Introduction to selenium webdriver
* Setting up Eclipse
* Downloading and configuring Webdriver in Eclipse
* Converting Selenium IDE test to programming language (JAVA)

**SeleniumDriver – IDE – JAVA – Junit**

* Junit Annotations
* Testing Annotations
* Testing Eclipse plugin
* Thread.Sleep
* Exceptions
* Webdriver Interface

**Concepts of Framework**

* Introduction to framework
* About framework
* Use of framework
* Different types of frameworks
* Different types of packages in framework

**DEVOPS COURSE CONTENT**

**LINUX Basics:**

* Unix and linux difference
* Linux File system structure
* Basic linux/unix commands
* Changing file permissions and ownership
* Types of links soft and hard link
* Filter commands
* Simple filter and advance filter commands
* Start and stop services
* Find and kill the process with id and name
* Package installation using RPM and YUM

**Introduction to Devops**

* Define Devops
* What is Devops
* SDLC models,Lean,ITIL,Agile
* Why Devops?
* History of Devops
* Devops Stakeholders
* Devops Goals
* Important terminology
* Devops perspective
* Devops and Agile
* Devops Tools
* Configuration management
* Continuous Integration and Deployment

**Introduction to Cloud computing**

* What is cloud computing
* Characteristics of cloud computing
* Cloud implementation models
* Cloud service models
* Advantages of cloud computing
* Concerns of cloud computing

**GIT: Version Control**

* Introduction
* What is Git
* About Version Control System and Types
* Difference between CVCS and DVCS
* A short history of GIT
* GIT Basics
* GIT Command Line
* Installing Git
* Installing on Linux
* Installing on Windows
* Initial setup
* Git Essentials
* Creating repository
* Cloning, check-in and committing
* Fetch pull and remote
* Branching
* Creating the Branches, switching the branches, merging the branches

**AWS**

* Creating AWS account
* Free tier Eligible services
* Understanding AWS Regions and availability zones
* EC2 ( Elastic Cloud Comput)
* About EC2 and types , Pricing
* EIP ( Elastic IP address), Allocating, associating , releasing
* Launch windows and Linux Instances in AWS
* Connecting windows and Linux instances from windows desktop and Linux machines
* S3 ( Simple Storage Service)
* About AWS Storage services, EBS and S3
* Creating S3 Buckets and putting objects in bucket
* Discussion about Bucket Properties
* S3 Pricing
* About S3 glecier
* EBS ( Elastic Block Storage)
* Types of EBS Volumes
* Creation, attaching and Detaching volumes
* ELB ( Elastic Load Balancer)
* Understanding the load balancing
* Configuring ELB and adding the webservers under ELB
* Auto Scaling
* Types of Scaling ( Horizontal and Vertical)
* Configuring Launch Configuration
* Creating and defining the auto scaling group policy
* IAM ( Identity Access Management)
* Understanding of AWS Security using IAM
* Definition of Roles, policies and Groups
* Creating IAM Users and managing password policies
* RDS ( Relational Database server)
* About RDS and available RDS Engines in AWS
* Configuring MYSQL RDS service
* Connecting EC2 Instance to RDS Instance

**Jenkins – Continuous Integration**

* Introduction.
* Understanding continuous integration
* Introduction about Jenkins
* Build Cycle
* Jenkins Architecture
* Installation
* Obtaining and installing Jenkins
* Installing and configuring Jenkins using WAR and RPM
* Java installation and configuration
* Maven Installation
* Exploring Jenkins Dashboard.
* Jobs
* Creating Jobs
* Running the Jobs
* Setting up the global environments for Jobs
* Adding and updating Plugins
* Disabling and deleting jobs
* Build Deployments
* Understanding Deployment.
* Tomcat installation and configuration

**Python**

Python Introduction

* what’s Python?
* Why do people use Python?
* Some quotable quotes
* A Python history lesson
* Advocacy News
* What’s Python good for?
* What’s Python not good for?
* The features list
* Python portability
* Summary

Using the Interpreter

* Python's Interactive Prompt
* Scripting
* Program Execution Model
* Program Architecture: modules
* How to run Python programs
* The IDLE interface
* Other python IDEs

Python Scripting

* Python Scripts in Linux/Unix & Windows
* Whitespace Significance
* Line Termination
* Comments in Python
* Basic Output Generation
* Simple User Input
* Python Modules
* Module Search Paths
* Determining the System Search Path
* input()
* raw\_input()

Working with Variables in Python

* Python Variables
* Naming Conventions & Rules
* Types as Objects
* Variable References & Garbage Collection
* Sequence Types
* Membership Statements
* List Iteration
* Sequence Assignments
* Mutable vs Immutable Objects
* Multi Target Assignments

Numeric Operations in Python

* More About Python's Numeric Types
* Numeric Tools
* The Decimal Module
* Operator
* Arithmetic
* Logical
* Relational
* Bitwise
* Special Operators
* Operator Precedence

Python Compound Statements

* Python Nesting Recap
* Comparison Operations
* The if Statement
* The if Ternary Expression
* The while Loop
* The for Loop
* Traversing Parallel Sets

Classes and Objects

* Introduction to OOP using python
* Classes and class attributes
* Instances and instance attributes
* Binding and method invocation
* Composition, Subclassing and Derivation
* Inheritance
* Built-in functions for classes, instances and other objects
* Privacy and Delegation
* An overview of built-in python classes and modules

Python String Types

* Generating Strings in Python
* Immutable
* Common String Methods
* Type Conversion in Python
* Formatting String Output
* Format Specifier
* Variable Substitution
* String Indexing
* String Slicing
* String Iteration

Python's Tuples

* Immutable
* Common Tuples Methods
* Tuples Operations
* Tuples Indexing
* Tuples Slicing
* Tuples Iteration
* Multi-Dimensional Tuples (Matrices)

Python's Lists

* Common List Methods
* The range() Function
* List Operations
* String Indexing
* String Slicing
* String Iteration
* Multi-Dimensional Lists (Matrices)

Python List Comprehension

* Basic List Comprehensions
* Compound List Comprehensions

Python Dictionaries

* Python Dictionaries
* Assigning Values to Dictionaries
* Dictionary Methods
* Dictionaries vs Lists & Tuples
* Dictionary Indexing
* Dictionary Iteration

Basic Input/Output with Files

* Opening Files
* Working with Files
* Controlling Output Location

Handling Compound Data Structure

* Nested Structure
* Parsing and Loading into Data Structure
* Retrival of data from the Data Structure

Creating Python Functions

* Function Basics
* Defining Functions
* Function Polymorphism
* Argument Defaults
* Lambdas
* Local Variables
* Understanding \_\_builtin\_\_
* Preventing Variable Modifications
* Argument Matching Methods
* Keyword Argument Methods

Regular Expression in Python

* Meta Characters
* re module
* Search
* Match
* Split
* Translation

Modules & Packages

* Module Basics
* Packages
* Using \_\_all\_\_ and \_ Variables
* Using \_\_name\_\_
* Using third party modules

Exceptions

* About Exceptions
* Python's Default Exception Handler
* Using Try/Except/Else/Finally Exceptions
* Generating User Defined Exceptions
* More on Exceptions
* Exception Examples
* Using Asserts
* Exception Classes