**Classroom Training Content for Campus Hires**

*A specialized content on Java & Web Technologies for SapientNitro*

# Some facts on the Training Program

## Recipients

The attendees and recipients of the Training Program are fresh College Graduates or Interns from renowned Engineering Colleges of India; with no professional experience.

## Consumers

The practices where the trainees would get consumed into after their classroom training are typically Commerce & Content or with SI where there is consulting needs for Java developers.

## Limitations

The training typically covers the pre-requisites before they get trained and start working on SCG specific technologies by the respective teams once they hit the Shadow Training Program.

Contents

[Some facts on the Training Program 2](#_Toc436639232)

[Recipients 2](#_Toc436639233)

[Consumers 2](#_Toc436639234)

[Limitations 2](#_Toc436639235)

[Classroom Training Structure 8](#_Toc436639236)

[Day Wise Classroom Program 10](#_Toc436639237)

[Day 1 – OOAD & UML 11](#_Toc436639238)

[Object Oriented Analysis and Design 11](#_Toc436639239)

[Fundamentals of UML 11](#_Toc436639240)

[UML Diagrams 11](#_Toc436639241)

[*Suggested Self-Study topics in UML* 11](#_Toc436639242)

[Day 2 [Half Day] - Working with Git 12](#_Toc436639243)

[Git (Configuration Management) 12](#_Toc436639244)

[Mapping Git with Eclipse 12](#_Toc436639245)

[Day 2 [Half Day] – RDBMS Concepts 12](#_Toc436639246)

[Introduction to RDBMS 12](#_Toc436639247)

[Introduction to SQL 12](#_Toc436639248)

[Day 3 – RDBMS Concepts 13](#_Toc436639249)

[Stored Procedures 13](#_Toc436639250)

[Database Views 13](#_Toc436639251)

[Database Triggers 13](#_Toc436639252)

[Day 4 – Java 14](#_Toc436639253)

[Introduction to Java 14](#_Toc436639254)

[Class Design 14](#_Toc436639255)

[Quality tools & Usage 14](#_Toc436639256)

[Day 5 – Java continued 15](#_Toc436639257)

[Polymorphism 15](#_Toc436639258)

[Inheritance 15](#_Toc436639259)

[Advanced Class Features 15](#_Toc436639260)

[Day 6 – Java continued 16](#_Toc436639261)

[Abstraction 16](#_Toc436639262)

[Encapsulation 16](#_Toc436639263)

[Java Bean 16](#_Toc436639264)

[Exception Handling 16](#_Toc436639265)

[Day 7 - Hands-On Practice Session - 01 17](#_Toc436639266)

[Hands-On Practice 17](#_Toc436639267)

[Day 8 – Java continued 18](#_Toc436639268)

[Exception Handling continued 18](#_Toc436639269)

[Threads 18](#_Toc436639270)

[Day 9 – Java Lang & IO Packages 19](#_Toc436639271)

[java.lang 19](#_Toc436639272)

[java.io 19](#_Toc436639273)

[Day 10 – Java Util Package 20](#_Toc436639274)

[java.util 20](#_Toc436639275)

[Java 8 Features 20](#_Toc436639276)

[Day 11 – JDBC (Java SQL Package) 21](#_Toc436639277)

[JDBC 21](#_Toc436639278)

[*Suggested Self-Study topics in Java* 21](#_Toc436639279)

[Day 12 - Hands-On Practice Session - 01 22](#_Toc436639280)

[Hands-On Practice 22](#_Toc436639281)

[Day 13 – Unit Testing with JUnit 23](#_Toc436639282)

[Introduction to Testing 23](#_Toc436639283)

[Working with JUnit 4 23](#_Toc436639284)

[Working with JUnit 4 continued... 23](#_Toc436639285)

[*Suggested Self-Study topics in JUnit* 23](#_Toc436639286)

[Day 14 – Effective Unit Testing 24](#_Toc436639287)

[Effective Unit Testing 24](#_Toc436639288)

[Day 15 – Ant & log4j 25](#_Toc436639289)

[Introduction to ANT 25](#_Toc436639290)

[Introduction to Log4j 25](#_Toc436639291)

[*Suggested Self-Study Topics in Log4j* 25](#_Toc436639292)

[Day 16 - Software Testing Basics 26](#_Toc436639293)

[Overview of Software Testing 26](#_Toc436639294)

[Types of Testing 26](#_Toc436639295)

[Methods of Testing 26](#_Toc436639296)

[Levels of Testing 26](#_Toc436639297)

[Day 17 & 18 – Classroom Project - 01 27](#_Toc436639298)

[Classroom Project 27](#_Toc436639299)

[Day 19 – Design Patterns & Principles 28](#_Toc436639300)

[Introduction to design patterns 28](#_Toc436639301)

[Creational Design Patterns 28](#_Toc436639302)

[Structural Design Patterns 28](#_Toc436639303)

[Day 20 – Design Patterns Principles continued... 29](#_Toc436639304)

[Behavioral Design Patterns 29](#_Toc436639305)

[*Suggested Self-Study topics in Design Patterns* 29](#_Toc436639306)

[SOLID Design Principles 29](#_Toc436639307)

[Day 21 – HTML/HTML5 Basics 30](#_Toc436639308)

[Introduction to HTML 30](#_Toc436639309)

[Formatting tags 30](#_Toc436639310)

[Links 30](#_Toc436639311)

[List 30](#_Toc436639312)

[Tables 30](#_Toc436639313)

[Forms 30](#_Toc436639314)

[Introduction to HTML5 30](#_Toc436639315)

[Features in HTML5 30](#_Toc436639316)

[HTML5 Forms 31](#_Toc436639317)

[Cascading Style Sheet (CSS) 31](#_Toc436639318)

[*Suggested Self-Study topics in HTML/CSS* 31](#_Toc436639319)

[Day 22 – JavaScript Fundamentals 32](#_Toc436639320)

[Introduction to JavaScript 32](#_Toc436639321)

[JavaScript deep dive 32](#_Toc436639322)

[Objects 32](#_Toc436639323)

[Validating form using DOM 32](#_Toc436639324)

[Event Handling 32](#_Toc436639325)

[Cookies 32](#_Toc436639326)

[*Suggested Self-Study topics in JavaScript* 32](#_Toc436639327)

[Day 23 – Script Frameworks 33](#_Toc436639328)

[Introduction to AJAX & JQuery 33](#_Toc436639329)

[AngularJs Basics 33](#_Toc436639330)

[Building blocks of AngularJs 33](#_Toc436639331)

[Day 24 – Basics of XML & JSON 34](#_Toc436639332)

[Overview of Extendible Markup Language (XML) 34](#_Toc436639333)

[JSON 34](#_Toc436639334)

[JSON Syntax 34](#_Toc436639335)

[Working with JSON 34](#_Toc436639336)

[*Suggested Self-Study topics in JSON* 34](#_Toc436639337)

[Day 25 – Hands-On Practice Session - 02 35](#_Toc436639338)

[Hands-On Practice Session 35](#_Toc436639339)

[Day 26 – Software Architecture & Java EE 36](#_Toc436639340)

[Overview of different Architectures 36](#_Toc436639341)

[Overview of JEE technologies 36](#_Toc436639342)

[Day 27 – JEE continued... 37](#_Toc436639343)

[Introduction to Servlet API 37](#_Toc436639344)

[Servlet Context and ServletConfig Interface 37](#_Toc436639345)

[Day 28 – Servlets, Filters and Listeners 38](#_Toc436639346)

[Session tracking in Servlets 38](#_Toc436639347)

[Working with Listeners 38](#_Toc436639348)

[Overview of Filters 38](#_Toc436639349)

[Day 29 – Java Server Pages 39](#_Toc436639350)

[Developing JavaServer Pages 39](#_Toc436639351)

[Day 30 – JSP continued... 40](#_Toc436639352)

[Tags in JSP 40](#_Toc436639353)

[*Suggested Self-Study topics in JSP* 40](#_Toc436639354)

[Day 31 & 32 – Classroom Project - 02 41](#_Toc436639355)

[Classroom Project 41](#_Toc436639356)

[Day 33 – Web Services & REST 42](#_Toc436639357)

[Introduction to web services 42](#_Toc436639358)

[Introduction to REST 42](#_Toc436639359)

[Day 34 – Spring IOC 43](#_Toc436639360)

[Introduction 43](#_Toc436639361)

[Spring Architecture 43](#_Toc436639362)

[Spring Core Container and Beans 43](#_Toc436639363)

[What are Setter Based and Constructor Based DI? 43](#_Toc436639364)

[XML Based Configuration 43](#_Toc436639365)

[Day 35 – Spring IOC continued... 44](#_Toc436639366)

[Bean LifeCycle 44](#_Toc436639367)

[Annotation Driven Configuration 44](#_Toc436639368)

[Database Access with Spring 44](#_Toc436639369)

[Day 36 – Spring MVC 45](#_Toc436639370)

[Introduction to Spring Web MVC 45](#_Toc436639371)

[Spring MVC Architecture 45](#_Toc436639372)

[Creating a Spring MVC application using Ant 45](#_Toc436639373)

[Day 37 – Spring MVC continued 46](#_Toc436639374)

[Validation Framework 46](#_Toc436639375)

[Integration of Spring MVC with JDBC and Hibernate 46](#_Toc436639376)

[Logging with Log4j 46](#_Toc436639377)

[Day 38 - Spring Boot 47](#_Toc436639378)

[Spring Boot Overview 47](#_Toc436639379)

[Creating a Spring Boot Application as a REST Service 47](#_Toc436639380)

[Day 39 - Spring Boot continued... 48](#_Toc436639381)

[Spring Boot Features 48](#_Toc436639382)

[Consuming a REST Service 48](#_Toc436639383)

[Day 40 – Hibernate Basics 49](#_Toc436639384)

[Introduction to ORM framework 49](#_Toc436639385)

[Hibernate Architecture Overview 49](#_Toc436639386)

[Database Connection and Schema Generation 49](#_Toc436639387)

[Persisting objects using Hibernate 49](#_Toc436639388)

[Day 41 – Hibernate Basics continued... 50](#_Toc436639389)

[CRUD Examples 50](#_Toc436639390)

[Lifecycle 50](#_Toc436639391)

[Setting up all types of Mapping 50](#_Toc436639392)

[Day 42 & 43 – Classroom Project - 03 51](#_Toc436639393)

[Classroom Project 51](#_Toc436639394)

[Day 44 - JCR & Jackrabbit 52](#_Toc436639395)

[Understanding Content 52](#_Toc436639396)

[Java Content Repository 52](#_Toc436639397)

[Apache Jackrabbit 52](#_Toc436639398)

[Day 45 - Modularity & OSGi 53](#_Toc436639399)

[Modularity Concepts 53](#_Toc436639400)

[Introduction to OSGi 53](#_Toc436639401)

[OSGi Bundles 53](#_Toc436639402)

[OSGi Services 53](#_Toc436639403)

[OSGi and Components 53](#_Toc436639404)

[Day 46 [Half-day] - NoSQL databases 54](#_Toc436639405)

[NoSQL Database 54](#_Toc436639406)

[Document Oriented Database 54](#_Toc436639407)

[*Suggested Self-Study topics in NoSQL databases* 54](#_Toc436639408)

[Day 46 [Half-day] - Server Side Scripting Basics 54](#_Toc436639409)

[Server-side Scripting 54](#_Toc436639410)

[Server-side Scripting in Java 54](#_Toc436639411)

[Server-side JavaScript 54](#_Toc436639412)

[Day 47 - Node.js 55](#_Toc436639413)

[What is Node.js? 55](#_Toc436639414)

[JavaScript 55](#_Toc436639415)

[Understanding Node.js Modules and Packages 55](#_Toc436639416)

[Reading and Writing Files 55](#_Toc436639417)

[Exploring Node.js Frameworks (Overview only) 55](#_Toc436639418)

[Working with Promises 55](#_Toc436639419)

[Working with Generator Functions 55](#_Toc436639420)

[Day 48 – Hands-On Practice Session – 04 56](#_Toc436639421)

[Hands-On Practice Session 56](#_Toc436639422)

# Classroom Training Structure

The classroom training is structured into six basic categories spanned across the training cycle:

* **Core Java & Unit Testing** 
  + Core Java
  + Junit
  + Effective Unit Testing
  + Software Testing Basics
* **Enterprise Frameworks** 
  + Git
  + RDBMS Basics
  + Ant & Log4j
* **Software Design Basics** 
  + OOAD & UML
  + Design Patterns
  + SOLID Design Principles
* **Ace the Web** 
  + HTML & CSS
  + JavaScript
  + Script Frameworks
  + XML & JSON
* **Web Architectures** 
  + JEE
  + JSP
  + Web Services & REST
  + Hibernate
  + Spring IOC & MVC
  + Spring Boot
* **Understanding the Content** 
  + JCR & Jackrabbit
  + Modularity & OSGi
  + Server-side Scripting Basics
  + NoSQL Databases
  + Node.js

Hands-on Practice sessions have been introduced to help the trainees practice the concepts learned during the classroom sessions.

Internal Projects have been introduced to help the trainees to learn working as a group under the supervision of the instructor.

# Day Wise Classroom Program

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Accelerators' Program - Java TOC - 2016** | | | | |
| Day 01 | OOAD & UML |  | Day 25 | Hands-on Practice Session |
| Day 02 | Git & RDBMS Basics |  | Day 26 | JEE |
| Day 03 | RDBMS Basics |  | Day 27 | JEE |
| Day 04 | Java |  | Day 28 | JEE |
| Day 05 | Java |  | Day 29 | Java Server Pages |
| Day 06 | Java |  | Day 30 | Java Server Pages |
| Day 07 | Hands-on Practice Session |  | Day 31 | Classroom Project 02 |
| Day 08 | Java |  | Day 32 | Classroom Project 02 |
| Day 09 | Java |  | Day 33 | Web Services & REST |
| Day 10 | Java |  | Day 34 | Spring IOC |
| Day 11 | Java |  | Day 35 | Spring IOC |
| Day 12 | Hands-on Practice Session |  | Day 36 | Spring MVC |
| Day 13 | Unit Testing with Junit |  | Day 37 | Spring MVC |
| Day 14 | Effective Unit Testing |  | Day 38 | Spring Boot |
| Day 15 | Software Testing |  | Day 39 | Spring Boot |
| Day 16 | Ant & Log4j |  | Day 40 | Hibernate Basics |
| Day 17 | Classroom Project 01 |  | Day 41 | Hibernate Basics |
| Day 18 | Classroom Project 01 |  | Day 42 | Classroom Project 03 |
| Day 19 | Design Patterns & Principles |  | Day 43 | Classroom Project 03 |
| Day 20 | Design Patterns & Principles |  | Day 44 | JCR & Jackrabbit |
| Day 21 | HTML/HTML5 Basics |  | Day 45 | Modularity & OSGi |
| Day 22 | JavaScript Fundamentals |  | Day 46 | No SQL DB Basics & Server Side Scripting |
| Day 23 | Script Frameworks |  | Day 47 | Node.js |
| Day 24 | XML & JSON |  | Day 48 | Hands-on Practice Session |

## Day 1 – OOAD & UML

### Object Oriented Analysis and Design

* Introduction to Object Oriented Analysis & Design (OOAD )
  + Object Oriented Methodology
  + Object Oriented Analysis & its primary tasks
  + Object oriented Design
  + Object oriented Modeling & its benefits
* Problem Statement
* Problem Specifications
* Identifying Object Classes
* Classes and Objects
* Prepare a Data Dictionary
* Identifying Association
* Identifying Attributes
* Refining with Inheritance
* Testing Access Path
* Iterating Object Modeling

### Fundamentals of UML

* Introduction to UML
* Goals of UML
* Use Case Modeling
* Static View
* Objects, Classes and the UML

### UML Diagrams

* Star UML
* Structure Diagrams
  + Class and Object Diagrams
  + Component and Deployment Diagram
* Behavior Diagrams
  + Use Case specifications
  + Interaction Diagrams
  + Sequence and Collaboration diagrams
  + Activity Diagram
* Associations (Composition and Aggregation)
* Relationships and Multiplicity

### *Suggested Self-Study topics in UML*

* Composite Structure diagram
* Package Diagram
* State Machine Diagram

## Day 2 [Half Day] - Working with Git

### Git (Configuration Management)

* Basics of Software Configuration Management
* Basics of Version Control System
  + Centralized version control system
  + Distributed/Decentralized version control system
* How to install, configure and create repository
* Working copy and check out
* Review, Commit & Update
* How to merge code and avoid conflict

### Mapping Git with Eclipse

## Day 2 [Half Day] – RDBMS Concepts

### Introduction to RDBMS

* Overview of Database Models
* ER Diagram and Normalization
* Features of Oracle

### Introduction to SQL

* Oracle Data types
* Introduction to DQL, DDL, DML, DCL
* CRUD operations with database
  + INSERT/UPDATE/DELETE/READ
* Working with Keys
  + Primary and Composite keys
  + Foreign Keys
* Use of Aggregate functions and ordering the result
  + Average(), Count(), Maximum(), Median(), Minimum(), Mode(), Sum()
  + Group By, Order By, Having
* Use of **LIKE** & **WHERE** clause
* Dropping / truncating a table
* Querying Multiple tables using joins
* SQL Joins
  + Inner, Outer and Self joins
* Nested queries

## Day 3 – RDBMS Concepts

### Stored Procedures

* Introduction to Stored Procedures
* Creating and executing Stored Procedures

### Database Views

* Introduction to views
* Scenarios when views are used
* Creating a view
* Inserting, deleting, updating data through a view

### Database Triggers

* Introduction to triggers
* Scenarios when Triggers are used
* Types of triggers
* Statement Level and Row Level triggers (Concepts only)
* Creating and Using triggers with tables

## Day 4 – Java

### Introduction to Java

* Features of Java
* JDK, JRE and JVM
* OOPs
  + Class, Object, Attribute, method
  + Access Modifiers – Private, Public
  + Constructors
  + Java API
* Data types and Operators
* Arrays & Copy Arrays
  + Primitive, Class, Array Types
  + Elements
  + Copy arrays
* Control Flow statements
  + If, Switch, For, While, and Do
  + Enhanced For Loop
* Enums

### Class Design

* Class and Objects
* Instance Variables and Static Variables
* Methods and Constructors

### Quality tools & Usage

* Checkstyle
* Debugging in Eclipse
* PMD

## Day 5 – Java continued

### Polymorphism

* Method Overloading
* Constructor Overloading
* Use of “this**”** keyword
* Variable Argument Method (Varargs) in Java
* Varargs versus Overloading

### Inheritance

* Inheritance Basics
* Use of super keyword
* Overriding
* Runtime Polymorphism

### Advanced Class Features

* Static variables, Static block and methods
* Static Import

## Day 6 – Java continued

### Abstraction

* Abstract classes and methods
* Final classes and methods
* Interfaces

### Encapsulation

* Packages and Access Specifiers

### Java Bean

* Introduction to Java Bean?
* Uses of Java Bean

### Exception Handling

* Exception and Errors
* Checked and Unchecked Exception
* Handling exceptions using try,catch and finally
* Java7 multi catch

## Day 7 - Hands-On Practice Session - 01

### Hands-On Practice

* Revise the concepts taught between days 1 through 6.
* Reinforce the learnings through a day long coding exercise.

## Day 8 – Java continued

### Exception Handling continued

* Use of throw and throws
* Use of Rethrow in Java7
* Automatic Resource Management in Java7
* Create Custom Exceptions

### Threads

* Introduction to Threads
* Thread class and Runnable Interface
* Thread priorities
* Methods of thread class
  + start(), run(), sleep(), interrupted(), isAlive(), stop(), wait(), join(), setPriority() & getPriority()
* Lifecycle of a thread
* Use of sleep(), join() methods
* Synchronization
* Inter-thread Communication

## Day 9 – Java Lang & IO Packages

### java.lang

* Wrapper classes
* Autoboxing and Unboxing
* Object class(equals and hashcode)
* Understanding toString method
* String, String Buffer and String Builder
* Conversions(string to object and vice versa)
* Comparable(to be taught after learning **java.util** framework)

### java.io

* Byte Streams and Character Streams
* InputStream and OutputStream
* Reader and Writer classes
* Read from and write to file/console
* Serialization
* Channel based IO

## Day 10 – Java Util Package

### java.util

* Generics
* Collections Framework
* Legacy classes Vector and Hashtable
* List, Set
* Map
* Iterator, ListIterator and Enumeration
* Collections and Array classes
* Sorting and searching
* Comparator vs Comparable

### Java 8 Features

* Lambda Expressions
* Function Interfaces

## Day 11 – JDBC (Java SQL Package)

### JDBC

* JDBC overview
* Types of drivers
* java.sql interfaces Driver, Connection, Statement
* Loading a driver and establishing a connection using DriverManager
* Perform CRUD operations using JDBC interfaces
* Prepared Statement for precompiled queries
* Callable Statement for stored procedures
* Mapping SQL and Java Types

### *Suggested Self-Study topics in Java*

* Event Handling in Java
* Java.net
* Packaging
* JavaDoc

## Day 12 - Hands-On Practice Session - 01

### Hands-On Practice

* Revise the concepts taught between days 8 through 11.
* Reinforce the learnings through a day long coding exercise.

## Day 13 – Unit Testing with JUnit

### Introduction to Testing

* Why Testing
* Overview of JUnit’s role in Testing
* Unit Testing with JUnit
* Installation and setup

### Working with JUnit 4

* Use of setUp() and tearDown()
* JUnit Assertions
* Static imports in Eclipse
* Executing test cases

### Working with JUnit 4 continued...

* Using @Test in JUnit4
* @BeforeClass and @AfterClass annotations
* TestFixtures with @Before and @After
* Ignoring Test Cases using @Ignore
* Testing Exceptions
* Testing Isolations
* Combining Test Cases as TestSuite
* Running Test Cases from main method via runner class
* Generating reports as XML/HTML

*Note: Focus of the day should be on understanding on how to write Test Cases.*

### *Suggested Self-Study topics in JUnit*

* JUnitExtensions - DBUnit, SQLUnit, HTMLUnit, JUnitEE
* JUnitExtensions – HttpUnit, XMLUnit, JWebUnit, JExample
* Testing with Stubs & Mocks

## Day 14 – Effective Unit Testing

### Effective Unit Testing

* Introduction to the Test Pyramid
* How to write effective Unit Tests
* Unit Tests as Safety Net
* Benefits of writing Effective Tests and
  + it’s after effects on code quality
  + it’s after effects on refactor-ability
  + it’s after effects on maintainability
* Safety over coverage
* Exercise

## Day 15 – Ant & log4j

### Introduction to ANT

* Install and configure ANT
* Use of build.xml
* Creating targets and their dependencies
* Use of property files
* Compile, Create JAR, doc using ant
* Running the application using ANT

### Introduction to Log4j

* What is Log4j?
* Installation of Log4j
* Components of Log4j
* Creating Logger objects and setting levels
* Priority of different levels
* Learn different types of Appenders and Layout
* Use of properties file
* Adding Logger object to a Java project
* Use different layouts and appenders to get log details

### *Suggested Self-Study Topics in Log4j*

* Log4j - Logging in Database

## Day 16 - Software Testing Basics

### Overview of Software Testing

* Who Tests - Developer, Tester, Manager, End User
* Difference between Verification & Validation
* Quality Assurance, Quality Control & Testing

### Types of Testing

* Manual Testing
* Automation Testing

### Methods of Testing

* Black Box
* White Box
* Grey Box

### Levels of Testing

* Functional
* Unit
* Integration
* System
* Regression
* Acceptance
* Non Functional Testing
  + Performance Testing
  + Load Testing
  + Stress Testing
* Security Testing

## Day 17 & 18 – Classroom Project - 01

### Classroom Project

* Apply concepts learnt during days 01 to 16.
* Mini project to be created under supervision of the Classroom Instructor.

## Day 19 – Design Patterns & Principles

### Introduction to design patterns

* Why design pattern?
* Principles of Design patterns
* Explore GOF design patterns
* Learn about the java and JEE design patterns

### Creational Design Patterns

* What is creational design pattern?
* Factory
* Abstract Factory
* Singleton
* Prototype

### Structural Design Patterns

* Understand the use of structural design pattern
* Adaptor
* Bridge
* Composite
* Decorator
* Facade
* Proxy

## Day 20 – Design Patterns Principles continued...

### Behavioral Design Patterns

* Understand the characteristic of behavioral design pattern
* Strategy
* Command
* Iterator
* Observer

### *Suggested Self-Study topics in Design Patterns*

* JEE Design Patterns in detail

### SOLID Design Principles

* Introduction to the SOLID design principles
* Violations of SOLID design principles
* Introduction to Code Smells
* Exercise

## Day 21 – HTML/HTML5 Basics

### Introduction to HTML

* HTML elements & attributes
* Empty tags vs container tags
* HTML file structure
* <head> and <body > elements
* use of <script>,<style>,<meta><title> tags

### Formatting tags

* Formatting tags in HTML
* Use of <pre > tag
* Heading tags

### Links

* Creating links using <a> tag
* External vs internal links
* Email links

### List

* Ordered & Unordered List
* Definition List

### Tables

* Creating tables
* Adding border to tables
* Rowspan and colspan
* Use of <legend> and <fieldset>
* Dealing with Images in HTML

### Forms

* What is a form?
* Use of <form> tag
* Form elements
* Textfield, Radio buttons ,checkbox, Textarea, dropdown
* Submit & reset buttons

### Introduction to HTML5

* What is HTML5?
* HTML5 Markup
* Deprecated Elements And Attributes
* Browser Support

### Features in HTML5

* Doctype declaration and character encoding
* Semantics and markup
* New And Updated Elements
* Structural Elements
* New Attributes

### HTML5 Forms

* New Input Types
* New Attributes
* Using placeholder data
* Marking required fields
* Working with number inputs
* Using date picker
* Form Validation
* Browser Support

### Cascading Style Sheet (CSS)

* Introduction to css
* CSS Rules
* Selectors
* Internal, External, Inline Stylesheet using ID & Class

### *Suggested Self-Study topics in HTML/CSS*

* CSS Layouts

## Day 22 – JavaScript Fundamentals

### Introduction to JavaScript

* What is JavaScript?
* JavaScript and Events
* JavaScript from External Files

### JavaScript deep dive

* The HTML Document and JavaScript
* Syntax
* Debugging in JavaScript
* Data types, Variables, Operators
* Types Of operators : Number, String or Boolean
* Control Structures, Block, Loops, for - in
* Identifying different types of dialog box(alert, confirm, prompt)
* Arrays in JavaScript

### Objects

* What Are Objects?
* User-Defined Objects
* Manipulating Objects
* Working with String, Number and Boolean

### Validating form using DOM

### Event Handling

* Introduction to Event Handlers
* Event Handlers as JavaScript Methods
* Handling Window, Document, History, Navigator objects
* Using Inner HTML with DOM in div tags for event Handling

### Cookies

* What Are Cookies?
* Creating and Reading Cookies with JavaScript

### *Suggested Self-Study topics in JavaScript*

* Object Oriented JavaScript

## Day 23 – Script Frameworks

### Introduction to AJAX & JQuery

* Getting Started with AJAX
* The Purpose of Ajax
* Traditional Web Application
* Creating a simple web application using AJAX
* Overview of JQuery
* Downloading JQuery or using from CDN
* Use of $(document). ready function
* Creating a simple web application using AJAX and JQuery

### AngularJs Basics

* What is SPA (Single Page Architecture)?
* What is AngularJs?
* Features of AngularJs
* Two way binding in AngularJs

### Building blocks of AngularJs

* Use of Directives
  + Directives Overview
  + Creating Directives
  + Directive Definition Object
* Working with ng-app, ng-bind, ng-model, ng-init and ng-repeat
* Working with Expressions
* Use of Filters and scope
  + Filter Overview & Understanding filter Expressions
  + Building Custom Filters
* Understanding Controllers

## Day 24 – Basics of XML & JSON

### Overview of Extendible Markup Language (XML)

* Introduction to XML
* XML as a Data Transport Format
* XML Tree – Elements, Attributes, Attributes, Nodes
* Creating Well-formed XML
* What is XML Schema? (Basic Idea)

### JSON

* Introduction to JavaScript Object Notation (JSON)
* JSON vs XML

### JSON Syntax

* JSON Datatypes & Objects
* JSON Name/value pairs
* JSON Arrays
* JSON values, Objects & Files

### Working with JSON

* Converting JSON to JavaScript using parse() method
* Creating JSON text from JavaScript using stringify() method
* Using JSON in Java

### *Suggested Self-Study topics in JSON*

* JSON Schema

## Day 25 – Hands-On Practice Session - 02

### Hands-On Practice Session

* Revise the concepts taught between days 19 through 24.
* Reinforce the learnings through a day long coding exercise.

## Day 26 – Software Architecture & Java EE

### Overview of different Architectures

* Introduction to different architectures
* Layers vs tiers
* Use of presentation, business, resource tiers
* Multitier Architecture
* What is JEE Architecture?
* Why JEE?
* Different types of JEE Architecture
* Introduction to MVC Architecture

### Overview of JEE technologies

* Deep dive into JEE
* Understand where each technology fits in JEE Architecture
* What is war, jar and ear
* Application Server vs Webserver

## Day 27 – JEE continued...

### Introduction to Servlet API

* Structure of web application
* Request and Response Model
* Servlet Lifecycle
* Types of Servlets
* HTTPServletRequest Object.
* Methods to get form data
* Developing, Packaging and deploying web application

### Servlet Context and ServletConfig Interface

## Day 28 – Servlets, Filters and Listeners

### Session tracking in Servlets

* Working with HTTP Session object
* Tracking session using cookies
* Implementing Session Tracking

### Working with Listeners

* What are listeners
* Usage of Listeners in web application
* Lifecycle and Attribute listeners
* Creating Context and Session lifecycle listeners

### Overview of Filters

* Introduction to Filters
* Lifecycle of Filters
* Use of doFilter Method
* Creating Web Applications with filters

## Day 29 – Java Server Pages

### Developing JavaServer Pages

* Introduction to JSP
* JSP lifecycle
* JSP Scripting Elements
* Difference between <%@include> and <jsp:useBean>
* JSP Implicit Objects
* Handling error in JSP pages
* Working with JSP:UseBean

## Day 30 – JSP continued...

### Tags in JSP

* Working with JSTL tags
* Core Tags in detail
* Function and formatting tags fundamentals
* Creating Java-based/JSP based Custom Tags
* Expressions Language (EL)

### *Suggested Self-Study topics in JSP*

* Database Tags & XML Tags
* Formatting & Function tags in detail

## Day 31 & 32 – Classroom Project - 02

### Classroom Project

* Apply concepts learnt during days 01 to 30.
* Mini project to be created under supervision of the Classroom Instructor.

## Day 33 – Web Services & REST

### Introduction to web services

* What are web services?
* Service-oriented Architecture
* Architecture of web services
* Types of web services -BIG and REST

### Introduction to REST

* Introduction to REST
* REST Architecture
* HTTP Methods
* Producing and Consuming a restful web service
* Use of @Path,@QueryParam, @PathParam
* Invoking a restful web service
* Invoking a RESTful Web-Service using Spring

## Day 34 – Spring IOC

### Introduction

* Overview of Spring framework
* Dependency Injection (DI)
* What is Spring?

### Spring Architecture

* Spring Container, IoC, DI
* Configuration Metadata – XML (structure only)
* Dependency Injection (DI) in Spring, DI Configuration

### Spring Core Container and Beans

* BeanFactory Interface and ApplicationContext
* ClassPathXmlApplicationContext, FileSystemXmlApplicationContext
* Spring Beans
* Loading Spring Configuration File

### What are Setter Based and Constructor Based DI?

### XML Based Configuration

* Setter Based and Constructor Based
* Factory Methods, Definition Inheritance (Parent Beans)
* Collection Valued Properties
  + Configuring and using Array, List, Map, Set and Properties
* Inner Bean
* Bean Scopes
* Autowiring of Dependencies using byName, byType, constructor

## Day 35 – Spring IOC continued...

### Bean LifeCycle

* InitializingBean and DisposableBean
* Creating custom lifecycle methods

### Annotation Driven Configuration

* Setter-Based /Constructor based
* Use of @Component, @Value
* Autowiring using @Autowired,@Qualifier
* Use of @PostConstruct and @PreDestroy

### Database Access with Spring

* Issues with JDBC / Typical JDBC Flow
* Introduction to Spring DAO Support
* Queries and Inserts
* Additional API Capabilities

## Day 36 – Spring MVC

### Introduction to Spring Web MVC

### Spring MVC Architecture

* DispatcherServlet and Context Loader Listener
* Use of Controllers, View Resolvers
* Stereotypes: @Component, @Service, @Controller, @Repository
* Handling Exceptions: HandlerExceptionResolver; @ExceptionHandler; Handling Standard Spring MVC Exceptions

### Creating a Spring MVC application using Ant

## Day 37 – Spring MVC continued

### Validation Framework

* Declarative Validation, @NotNull, @Size, @Min, etc
* Custom Validators

### Integration of Spring MVC with JDBC and Hibernate

* Internationalization
* Spring Tags

### Logging with Log4j

## Day 38 - Spring Boot

### Spring Boot Overview

* Overview of Spring Boot
* Installation and System Requirements
* Building with Maven
  + Introduction to Maven & its objectives
  + Install and setting up Maven
  + Maven project structure
  + POM.xml description
* Spring Boot Starter packages
* Packaging as a WAR or JAR

### Creating a Spring Boot Application as a REST Service

* Structuring the code
* Using @EnableAutoConfiguration, @ComponentScan, @Configuration
* Use of @SpringBootAppilcation annotation
* Packaging as a Jar
* Packaging as War and deploying in external Tomcat

## Day 39 - Spring Boot continued...

### Spring Boot Features

* SpringApplication
* Externalized Configuration
* Profiles
* Logging

### Consuming a REST Service

* Consume Spring REST Service from spring MVC

## Day 40 – Hibernate Basics

### Introduction to ORM framework

* ORM & ORM Frameworks
* JDBC Vs Hibernate
* Features of Hibernate

### Hibernate Architecture Overview

* Configuration and Session Factory
* Session
* Transaction
* Environment Setup
* Getting started with Hibernate quickly

### Database Connection and Schema Generation

* Mapping JavaBeans to tables
* Mapping properties to columns
* Building a Configuration Programmatically /Using .properties file /Using XML configuration file
* Creating a HBM document from the java bean class
* Creating a java bean class from a HBM document

### Persisting objects using Hibernate

## Day 41 – Hibernate Basics continued...

### CRUD Examples

* Persisting data
* Loading data into an object
* get Vs load
* Deleting, updating & finding objects

### Lifecycle

* Transient State
* Persistent state
* Detached state

### Setting up all types of Mapping

* Mapping composite keys
* Mapping a java class
* Mapping Association (one to one and one to many only)
* Component Mapping

## Day 42 & 43 – Classroom Project - 03

### Classroom Project

* Apply concepts learnt during days 01 to 41.
* Mini project to be created under supervision of the Classroom Instructor.

## Day 44 - JCR & Jackrabbit

### Understanding Content

* What is Content?
* Processes of Content Management
* Content Management Systems
  + Content management application
  + Content delivery application

### Java Content Repository

* What is JCR
* Need for JCR API
* Content Repository Model

### Apache Jackrabbit

* How Jackrabbit works
* Jackrabbit Architecture
* Configuring Jackrabbit
  + Repository Home Directory
  + Repository configuration file
* What is Object Content Mapping

## Day 45 - Modularity & OSGi

### Modularity Concepts

* What is Modularity?
* Modularity vs Object Orientation

### Introduction to OSGi

* What and Why of OSGi
* An architectural overview of OSGi
* Hello World Examples with Module, Lifecycle & Service Layers

### OSGi Bundles

* Introduction to bundles
* Basics of Bundle Life-cycle

### OSGi Services

* What, why & when of Services
* Concepts of Publishing a service
* Concepts of Finding & Binding Services

### OSGi and Components

* What & why of components
* Overview Service-oriented Component Model (Conceptual only)

## Day 46 [Half-day] - NoSQL databases

### NoSQL Database

* NoSQL - When & Where
* NoSQL - Pros & Cons
* Types of NoSQL Database (Concepts only)
  + Column, Document, Key-Value, Graph, Multi-Model

### Document Oriented Database

* Documents - Keys, Retrieval & Organization
* Comparison with Relational Databases

### *Suggested Self-Study topics in NoSQL databases*

* Concepts of MongoDB

## Day 46 [Half-day] - Server Side Scripting Basics

### Server-side Scripting

* What is Server-side Scripting?
* Server-side vs Client-side Scripting

### Server-side Scripting in Java

* Java Server Pages vs Scripting for the Java platform
* Sneak-peak into javax.script

### Server-side JavaScript

* Basic Concepts

## Day 47 - Node.js

### What is Node.js?

* Introducing Node.js
* Installing Node.js
* Understanding asynchronous tasks
* What are callbacks?

### JavaScript

* Why is Node.js written in JavaScript?
* JavaScript benefits and features

### Understanding Node.js Modules and Packages

* Writing your own module
* Managing third-party packages with npm
* What is the package.json file?

### Reading and Writing Files

* Reading from files
* Accessing directories
* Writing to files

### Exploring Node.js Frameworks (Overview only)

* What is a framework for Node.js?
* A look at the Express framework
* Serving static files with Express
* Creating an endpoint with Express
* A look at the Sails framework
* Creating an endpoint with Sails
* A look at the Koa framework
* Creating an endpoint with Koa

### Working with Promises

* What are promises?
* Dealing with nested callbacks
* Generating promises with the BlueBird library
* Creating a custom promise

### Working with Generator Functions

* What are generators?
* Callbacks versus generators

## Day 48 – Hands-On Practice Session – 04

### Hands-On Practice Session

* Revise the concepts taught between days 44 through 47.
* Reinforce the learnings through a day long coding exercise.