

SAPIENT
RAZORFISH_

Classroom Training Content for Campus Hires

A specialized content on Full Stack Java Technologies for SapienRazorfish

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
Recipients.....	2
Consumers.....	2
Limitations.....	2
Training Structure	7
Day Wise Classroom Program	8
Digital Academy Mandatory Pre-read Courses	9
Day 1 – OOAD & UML	9
Day 2 – RDBMS Concepts.....	9
Day 3 – Java (Version - 8)	10
Day 4 – Java continued.....	10
Day 5 – Java continued.....	11
Day 6 – Java continued.....	11
Day 7 – Java Inbuilt packages	12
Day 8 – Java Inbuilt packages	12
Day 9 – Java Inbuilt Packages	13
Day 10 – JDBC (Java SQL Package)	13
Day 01 – Refresher to Digital Academy Mandatory Pre-read	14
Refresher.....	14
Day 02 – GIT & Maven.....	15
Git (Configuration Management)	15
Introduction to Maven (version – 3.3.3)	15
Day 03 & 04 –Unit Testing Tools	16
Introduction to Testing	16
Working with JUnit 5	16
Mockito Framework (version – 2.2.26)	16
Introduction to SL4J (version - 1.7.23)	16
Day 05 – Design Patterns & Principles.....	17
Overview of Design Pattern	17
SOLID Design Principles.....	17
Day 06 – Hands-on Practice Session 01.....	18
Day 07 –HTML5 & CSS	19
Introduction to HTML5	19
Formatting tags, Links & Lists	19

Tables	19
Forms	19
Media Elements	19
Overview of canvas	19
Cascading Style Sheet (CSS)	19
Day 08 – JavaScript	20
Introduction to JavaScript	20
JavaScript deep dive	20
Event Handling	20
Form validation using DOM.....	20
Day 09 – JavaScript Objects, Basics of JSON, XML, AJAX	21
JavaScript Objects	21
Understanding DOM	21
Using Cookie	21
Webstorage API for HTML.....	21
JSON.....	21
Overview of XML.....	21
Overview of AJAX.....	21
Day 10, 11 & 12 – Java EE & JSP	22
Overview of different Architectures.....	22
Overview of JEE technologies.....	22
Introduction to Servlet API.....	22
Session tracking in Servlets.....	22
Developing Java Server Pages	22
Using Expression Language	22
Tags in JSP	22
Day 13 & 14 – REST & Swagger	23
Introduction to web services.....	23
Introduction to REST	23
Using Swagger API (Version – 2.0).....	23
Day 15 – Hibernate Basics.....	24
Introduction to ORM framework	24
Hibernate Architecture Overview	24
CRUD Examples.....	24
Lifecycle of Hibernate	24

Mapping Association (using annotations only)	24
Day 16, 17 & 18 – Spring IOC with SpringBoot.....	25
Introduction to Spring Framework	25
Spring Architecture	25
Creating a Spring Application using Spring Boot	25
What are Setter Based and Constructor Based DI?.....	25
Annotation Based Configuration	25
Bean LifeCycle.....	25
Java based Configuration	25
Database Access with Spring	25
Create a REST Application using SpringBoot.....	25
Day 19, 20 & 21 – Spring MVC.....	26
Introduction to Spring Web MVC	26
Spring MVC Architecture	26
Validation Framework.....	26
Integration of Spring MVC with Hibernate.....	26
Spring with JPA Repositories.....	26
Consuming a REST Service.....	26
Day 22, 23 & 24 – Spring Microservices	27
Introduction to Microservices	27
Microservices using Netflix Components	27
Day 25 – Hands-on Practice Session 02.....	28
Day 26 - MongoDB Basics	29
NoSQL Database.....	29
Introduction to MongoDB (version – 3.6).....	29
Querying in MongoDB.....	29
Day 27 & 28 - Node.js	30
What is Node.js?.....	30
Node Package Manager (npm) and Node.js modules	30
HTTP Server.....	30
Exploring Node.js Frameworks	30
Accessing a Node endpoint using a html page	30
Day 29 – Angular 2/4	31
Introduction to Angular 2/4	31
Working with Angular 2/4.....	31

Day 30 - JCR & Jackrabbit	32
Understanding Content	32
Java Content Repository	32
Apache Jackrabbit	32
Day 31 - Modularity & OSGi	33
Modularity Concepts	33
Introduction to OSGi	33
OSGi Bundles	33
OSGi Services	33
OSGi and Components	33
Day 32 – Hands-on Practice Session 03	34

Training Structure

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

- **Core Java & Unit Testing**
 - Core Java *
 - Junit & Mockito **
- **Enterprise Frameworks**
 - Git
 - RDBMS Basics *
 - Maven & SL4J
- **Software Design Basics**
 - OOAD & UML *
 - Design Patterns **
 - SOLID Design Principles **
 - Code Smells **
- **Ace the Web**
 - HTML & CSS **
 - JavaScript
 - XML, JSON & Ajax
- **Web Architectures**
 - JEE (Servlets & JSP)
 - REST & Swagger API
 - Hibernate
 - Spring IOC & MVC
 - Spring Boot
 - Spring Microservices
- **Understanding MEAN**
 - Mongo DB
 - Node
 - Angular
- **Understanding Content**
 - Modularity & OSGi
 - Jackrabbit

* Digital Academy Only

** Digital Academy with Compressed Classroom

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

Day Wise Classroom Program

Sapient Confidential

Digital Academy Mandatory Pre-read Courses

Note: The following 10 days of Digital Academy Pre-read Courses are NOT TO BE COVERED in classroom training. This is just for the trainer to understand the level they have already gone through before joining.

Day 1 – OOAD & UML

Object Oriented Analysis and Design

- Introduction to Object Oriented Analysis & Design (OOAD)
- Classes and Objects
- Object oriented concepts

Fundamentals of UML

- Introduction to UML
- Goals of UML
- Use Case Modeling

UML Diagrams

- Structure Diagrams
 - Class and Object Diagrams
 - Component and Deployment Diagram
- Behavior Diagrams
 - Use Case specifications
 - Interaction Diagrams - Sequence and Collaboration
 - Activity Diagram
- Associations (Composition and Aggregation)

Day 2 – RDBMS Concepts

Introduction to RDBMS

- Overview of Database Models
- Overview of ER Diagram and Normalization

Introduction to SQL

- Oracle Data types
- Introduction to DQL, DDL, DML, DCL
- CRUD operations with database
 - INSERT/UPDATE/DELETE/RETRIEVE
- Using 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
- SQL Joins - Inner, Outer and Self joins
- Sub queries – single row, multiple row, correlated

Database Views

- Introduction to views
- Scenarios when views are used

Day 3 – Java (Version - 8)

Introduction to Java

- Features of Java 8
- JDK, JRE and JVM
- OOPs
 - Class, Object, Attribute, method
 - Access Modifiers – Private, Public
- Data types and Operators
- Arrays
 - Primitive, Class, Array Types
 - Elements
- 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

Day 4 – Java continued

Polymorphism

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

Inheritance

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

Abstraction

- Abstract classes and methods
- Final classes and methods

Encapsulation

- Packages and Access Specifiers
- What is a java bean?

Day 5 – Java continued

Static

- Static variables, Static block and methods
- Static Import

Interfaces

- Understanding interfaces
- Extending Interfaces
- Functional Interfaces
- Default methods in interfaces

Lambda Expressions

- What is Lambda Expression?
- Syntax, structure and usage
- When to use Lambda?

Day 6 – Java continued

Exception Handling

- Exception and Errors
- Checked and Unchecked Exception
- Handling exceptions using try, catch and finally
- Using Multicatch
- Use of throw and throws
- Automatic Resource Management – Try with resources
- Creating Custom Exceptions

Multithreading Basics

- Overview of Multithreading
- Process vs Thread
- Main Thread, priorities
- Creating child threads – extending Thread
- Creating child threads – Implementing Runnable
- Synchronization
- Daemon Thread

Day 7 – Java Inbuilt packages

java.lang

- Wrapper classes
- Autoboxing and Unboxing
- Object class
- Shallow Cloning – overview and example
- Use of equals and hashCode
- Understanding toString() method
- String, String Buffer and String Builder

java.io

- Overview of Byte Streams and Character Streams
- Serialization

java.util

- Collections Framework
- List – ArrayList, LinkedList,
- Set – HashSet, LinkedHashSet, TreeSet
- Map – TreeMap, ConcurrentHashMap
- Iterator, ListIterator
- Sorting using Comparator and Comparable
- Method References – overview and example
- Properties, UUID, Random – overview and example
- Calendar, Locale – overview and example
- Regular Expressions

Day 8 – Java Inbuilt packages

java.util.stream

- What is a stream?
- Collections vs Streams
- Creating Streams
- Different types of streams
- Intermediate Operations on Streams
 - map(), filter(), sorted(), flatMap()
- Terminal Operations on Streams
 - reduce(), forEach(), findFirst(), match(), collect(), count()
- Converting a Stream to a collection or Array
- Numeric Streams – IntStream, DoubleStream, LongStream

java.util.concurrent

- Overview of concurrency API
- Executors
 - Executor Interfaces
 - Thread Pools
 - Using Callable and Future

Day 9 – Java Inbuilt Packages

java.nio

- Overview of Java NIO
- Channels, Buffers & Selectors
- Channel Implementations
 - FileChannel
- Usage of Buffer
 - Buffer Types
 - Allocating a buffer
 - Writing data to a buffer
 - Reading data from a buffer
- Scattered Reads & Gathering Writes
- Channel to Channel Transfers
- Usage of Selectors
 - Creating a selector
 - Registering channels with the selector
 - Selecting Channels via a selector
- Using AsynchronousFileChannel
 - Reading data through Future and Completion Handler
 - Writing data through Future and Completion Handler

java.time

- Why java Date/Time API
- LocalDate and LocalTime
- Time zones and time stamps
- Temporal Adjustments
- Period and Duration
- Formatting and parsing
- Conversion between different date/time objects

Day 10 – JDBC (Java SQL Package)

JDBC

- JDBC overview
- 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

Day 01 – Refresher to Digital Academy Mandatory Pre-read

Refresher

- OOAD & UML – 1 Hour
- RDBMS Basic – 2 Hours
- Core Java – 5 Hours

Note: Refresher should be limited to the level of doubt clarification/ question-answer sessions only. At the end of day 1, there would be an Assessment on what they would have already gone through in Digital Academy.

Day 02 – GIT & Maven

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

Introduction to Maven (version – 3.3.3)

- What is a build tool?
- Overview of Maven
- Environment Setup
- Maven POM File
- Creating a java project using maven
- Maven Directory Structure
- Project Dependencies
 - External Dependencies
 - Snapshot dependencies
- Maven Repository – Local, Central, Remote
- Build Lifecycle, phases and goals
- Executing the mvn command

Day 03 & 04 –Unit Testing Tools

Introduction to Testing

- Why Testing
- Overview of JUnit
- JUnit 5 Architecture
- Environment setup

Working with JUnit 5

- Creating Testcases
- JUnit 5 Annotations
- JUnit Assertions
- Assumptions
- Using @Test in JUnit5
- Using Annotations - @BeforeAll and @AfterAll
- TestFixtures with @BeforeEach and @AfterEach
- Testing Exceptions using assertThrows
- Combining Test Cases as TestSuite
- Using @RepeatedTest
- Tagging and filtering

Mockito Framework (version – 2.2.26)

- Overview of Mockito and mock objects
- Using Mockito API
- Adding mockito to a project
- Creating mock objects using @Mock Annotation
- Configuring mock objects
- Dependency Injection using @InjectMocks
- Adding behaviour
- Verify the behaviour using verify()
- Verify the call order
- Handling Exceptions
- Wrapping the java objects using @spy or spy()

Introduction to SL4J (version - 1.7.23)

- What is logging?
- Introduction to SL4J
- When to use SL4J?
- Binding with different logging frameworks
- Using SL4J with logback logger
- What is Logback?
- Architecture & Configuration
- Logger, Appenders and Layouts
- Parameterized logging
- Logging Separation

Day 05 – Design Patterns & Principles

Overview of Design Pattern

- Overview of creational, structural, behavioral, concurrency design pattern
- Example for Singleton Design Pattern only

SOLID Design Principles

- Introduction to the SOLID design principles
- Violations of SOLID design principles
- Introduction to Code Smells
- Refactoring techniques

Sapient Confidential

Day 06 – Hands-on Practice Session 01

Sapient Confidential

Day 07 –HTML5 & CSS

Introduction to HTML5

- Introduction of HTML5
- HTML file structure, elements, attributes
- <head> and <body> element
- Use of <script>, <style>, <meta><title> tags
- Doctype declaration

Formatting tags, Links & Lists

- Formatting and Heading tags
- External vs internal links - using <a> tag
- Adding Images to a webpage
- Ordered, Unordered & Description List

Tables

- Creating tables - <table>, <tr><thead><td>
- Use of <legend> and <fieldset>

Forms

- Use of <form> tag
- Form elements - Textfield, Radio button, checkbox, Textarea, dropdown
- Submit & Reset buttons
- New Form Elements – output, datalist
- New Input Types and attributes

Media Elements

- Audio and video tags

Overview of canvas

Cascading Style Sheet (CSS)

- Introduction to css
- CSS Rules & Selectors
- Internal, External, Inline Stylesheet
- Working with background, text, font
- Styling List and Link
- Using ID & Class
- Floating Elements using CSS
- CSS positioning
- Overview of BOX Model and transitions
- Cross Browser compatibility issues and solutions

Day 08 – JavaScript

Introduction to JavaScript

- Introduction to JavaScript
- Syntax and structure

JavaScript deep dive

- The HTML Document and JavaScript
- Data types, Variables, Operators
- Control Structures, Block, Loops, for - in
- Different types of dialog box (alert, confirm, prompt)
- Using Functions

Event Handling

- Introduction to Event Handlers
- Event Handlers as JavaScript Methods
 - Onclick, onload, onkeyup
- Using Inner HTML with DOM in div tags for event Handling

Form validation using DOM

Day 09 – JavaScript Objects, Basics of JSON, XML, AJAX

JavaScript Objects

- Objects in JavaScript
- Inbuilt Objects – Number, Boolean, String, Array, Date, Math
- User-defined Objects – Object Literals
- User-defined Objects – Using Function Constructor
- Properties, Methods of Objects
- Modifying and accessing Object's properties/methods

Understanding DOM

- Handling Window, Document, History, Navigator objects

Using Cookie

Webstorage API for HTML

- LocalStorage and SessionStorage

JSON

- Introduction to JavaScript Object Notation (JSON)
- JSON Datatypes & Objects
- Creating a JSON File
- Using parse() and Stringify() method
- Using JSON with Java – Encoding and Decoding

Overview of XML

- Introduction to XML
- Syntax & Structure
- What is well formed XML

Overview of AJAX

- Introduction to Ajax
- Simple workflow of Ajax
- Use of XMLHttpRequest Object

Day 10, 11 & 12 – Java EE & JSP

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

Introduction to Servlet API

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

Session tracking in Servlets

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

Developing Java Server Pages

- Introduction to JSP
- JSP lifecycle
- JSP Scripting Elements
- `<@ include>` and `<jsp:include>`
- JSP Implicit Objects
- Handling Exception in JSP pages

Using Expression Language

Tags in JSP

- Working with JSTL tags
- Core Tags in detail
- Example for Function and formatting tags example
- Creating Java-based/JSP based Custom Tags

Day 13 & 14 – REST & Swagger

Introduction to web services

- What are web services?
- Service-oriented Architecture
- Architecture of web services

Introduction to REST

- Introduction to REST
- REST Architecture
- HTTP Methods
- Producing and Consuming a restful web service
- Use of @Path,@QueryParam, @PathParam
- Caching

Using Swagger API (Version – 2.0)

- Introduction to Swagger
- Benefits of Swagger
- Swagger Tools
 - Swagger spec
 - Swagger editor
 - Swagger-UI
 - Swagger-codegen
- Auto-generating the Swagger file from code annotations
- Integrating REST with Swagger
- Creating REST API from Swagger Spec
- Generate REST client with Swagger Spec

Day 15 – Hibernate Basics

Introduction to ORM framework

- ORM & ORM Frameworks
- Features of Hibernate
- Installation and Environment Setup

Hibernate Architecture Overview

- Configuration and Session Factory
- Session
- Transaction

CRUD Examples

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

Lifecycle of Hibernate

Mapping Association (using annotations only)

- Mapping OneToOne
- Mapping OneToMany

Day 16, 17 & 18 – Spring IOC with SpringBoot

Introduction to Spring Framework

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

Spring Architecture

- Spring Container, IoC, DI
- Dependency Injection (DI) in Spring, DI Configuration
- BeanFactory Interface and ApplicationContext
- Spring Beans

Creating a Spring Application using Spring Boot

- Overview of Spring Boot | Installation and System Requirements
- Spring Boot Starter packages
- Using spring boot to create ready to run spring application
- Structuring the code
- Using @EnableAutoConfiguration, @ComponentScan, @Configuration
- Use of @SpringBootApplication annotation
- Packaging as a Jar

What are Setter Based and Constructor Based DI?

Annotation Based Configuration

- Setter Based and Constructor Based
- Factory Methods, Definition Inheritance (Parent Beans)
- Bean Inheritance
- Use of @Component, @Value
- Collection Valued Properties
 - Configuring and using Array, List, Map, Set and Properties
- Inner Bean
- Bean Scopes
- Autowiring of Dependencies using byName, byType, constructor
 - Autowiring using @Autowired, @Qualifier
- Use of @PostConstruct and @PreDestroy
- Using PropertyPlaceholder external property file

Bean LifeCycle

- InitializingBean and DisposableBean
- Creating custom lifecycle methods

Java based Configuration

- Understanding @Configuration
- Using @Bean
- Autowiring

Database Access with Spring

- Introduction to Spring DAO Support
- Queries and Inserts
- Additional API Capabilities

Create a REST Application using SpringBoot

Day 19, 20 & 21 – Spring MVC

Introduction to Spring Web MVC

Spring MVC Architecture

- MVC Architecture
- DispatcherServlet
- Creating a spring application using Spring Boot
- Use of Controllers, View Resolvers
- Stereotypes: @Component, @Service, @Controller, @Repository
- Handling Exceptions
 - HandlerExceptionResolver
 - Controller based ExceptionHandler
 - Global Exception Handler
- Handling Sessions

Validation Framework

- Spring Validation API
- JSR 303 Validation API

Integration of Spring MVC with Hibernate

Spring with JPA Repositories

Consuming a REST Service

- Consume Spring REST Service from Spring MVC

Day 22, 23 & 24 – Spring Microservices

Introduction to Microservices

- What is Microservices?
- Why Microservices?
- Characteristics of Microservices
- Patterns in Microservices Architecture

Microservices using Netflix Components

- What is Netflix OSS
- Components of Netflix oss
 - Netflix Eureka
 - Netflix Ribbon
 - Netflix Zuul

Spring Cloud and Netflix OSS

- Getting Started
- Creating Microservices using SpringBoot
- Using Netflix Eureka as Service Registry
- Routing and filtering
- Using the Netflix Zuul edge service library for routing and filtering.
- Fault Handling using Netflix Hystrix, Circuit Breaker

Day 25 – Hands-on Practice Session 02

Sapient Confidential

Day 26 - MongoDB Basics

NoSQL Database

- Introduction to NoSQL
- Types of NoSQL Database (Concepts only)
 - Column, Document, Key-Value, Graph, Multi-Model

Introduction to MongoDB (version – 3.6)

- Introduction to MongoDB
- Installation and set up
- Data Modelling and Architecture
- Advantages of MongoDB - Why & Where to use.

Querying in MongoDB

- Creating & Dropping Database & Collection
- Insert, Query, Update & Delete Document
- find(), limit(), skip(), sort(), ensureIndex() methods
- General concepts of Aggregation & Sharding

Integrating Java Application with MongoDB

- CRUD operation

Note: The contents for MongoDB is currently under discussion with Sapient SMEs. Subjected to change.

Day 27 & 28 - Node.js

What is Node.js?

- Introduction to Node.js
- Installation and configuration

Node Package Manager (npm) and Node.js modules

- The Node Package Manager
- Creating a project
- The package.json configuration file
- Global vs. local package installation
- Writing your own module
- Adding dependencies using npm

HTTP Server

- Building an HTTP server
- Rendering a response
- Processing query strings

Exploring Node.js Frameworks

- What is a framework for Node.js?
- Introduction to Express4 framework
- Using Express generator
- Serving static files with Express
- Creating an endpoint with Express
- Routing
- Writing and using middleware in NodeJs
- Templating using express-handlebars
- Building Restful APIs using Node and Express
- Reading Post data

Accessing a Node endpoint using a html page

Day 29 – Angular 2/4

Introduction to Angular 2/4

- Overview of Angular 2/4
- Features of Angular 2 /4
- Angular 2/4 Architecture

Working with Angular 2/4

- Understanding Typescript
- Modules and components
- Component Directives
- Structural Directives
- Attribute Directives
- Data Binding and Data Display in Angular 2/4
- User Input with Angular 2/4
- Binding to User Input Events

Note: Version of Angular, whether 2 or 4, which fits best for the fresh graduates, is currently under discussion with Sapient SMEs. Subjected to change.

Day 30 - 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 31 - 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 32 – Hands-on Practice Session 03

Sapient Confidential