ABOUT THE COURSE

COURSE NAME:	Java Spring Framework
TOTAL DURATION:	45 Hrs
MODE OF DELIVERY	PHYSICAL CLASSROOM TRAINING AT RESPECTIVE COLLEGES
TRAINER TO STUDENT RATIO:	1:50
TOTAL MARKS:	75

Table 1	
OVERALL COURSE OBJECTIVE:	Java and the Spring Framework are cornerstones of modern enterprise application development. For students aspiring to become full stack developers, this is a core component of the stack. Java is a robust, platform-independent language that remains highly popular in enterprise environments. Spring is the de facto standard framework for building Java applications, widely used in various industries. Learning Java with Spring provides students with in-demand skills that will make them industry ready. It will help them enhance their skills further to microservices based applications, cloud native applications and the like.

LEARNING OUTCOME:	After completing this course, the learners would be able to:- • Use optional class, date/Time API, Functional Interfaces, apply Lambda Expressions, Streams API • Use Maven for project building and repository • Determine the challenges of mapping objects to relational database and need of ORM (Object Relational Model) frameworks • Apply Hibernate mapping files and annotations to map Java classes and object associations to relational database

	tables using Hibernate APIs • Implement CRUD operations in Hibernate using Eclipse as the IDE • Develop application using Aspect Oriented Programming • Develop application with different configurations for multiple environments (DEV, PROD)using Spring Profiles

TABLE 2: MODULE WISE COURSE CONTENT AND OUTCOME				
SL. NO	**MODULE NAME**	**MODULE CONTENT**	**MODULE LEARNING OUTCOME**	**DURATION (HRS)**
1	Java SE 8 Features	Evolution of Java, Default and Static Methods, Lambda Expressions, Streams, Optional Class, Date Time API.	Write interfaces with the default and static methods. • Make use of Optional class. Work with Date/Time API Develop Functional Interfaces Apply Lambda Expressions Work with Streams API	9
2	Apache Maven	Why Maven?,What is Maven?, Getting Started with Maven, Maven Archetypes, Project Object Model – POM, Maven Repositories and Dependency Management, Life-cycle Phases of Maven, Maven Plugins and Goals.	Explain the features of Maven, structure of a Maven project, Maven repositories and life- cycle phases of Maven. • Create Project Object Model (POM) based on the project requirement. • Create Maven based Java standalone/web applications in	9

			Eclipse/IntelliJ IDE. • Build Java based applications using various goals and plugins of Maven • Perform quality testing on the code using SonarQube plugin • Generate code-coverage report using Cobertura plugin. • Create Javadoc for the application using reporting plugin – Javadoc. • Installing and working with Maven from command prompt	
3	Hibernate Framework – Basics	Data Persistence, Object Relational Impedance Mismatch, Introduction to Hibernate Framework, Scenario, Hibernate Architecture, Hibernate APIs, Hibernate Entity States, Identifier Generation Strategies, Association Mapping, Querying Mechanisms in Hibernate, Migrating to Hibernate Framework	• Determine the challenges of mapping objects to relational database and need of ORM(Object • Relational Model) frameworks. Describe Hibernate • Features and Hibernate Architecture. Apply Hibernate mapping files and annotations to map Java classes and object associations to relational database tables	9

			using Hibernate APIs. • Implement CRUD operations in • Hibernate using Eclipse as the IDE. • Differentiate the Identifier Generation Strategies. • Describe the concept of Association Mapping. • Understand Hibernate Entity States and implement them • Develop queries using Native SQL statements, Hibernate Query Language(HQL). • Application migration from Hibernate version 4 to 5 Hibernate Best Practices	
4	Spring 5 Basics with Spring Boot	Introduction to Spring Framework, Spring Inversion of Control, Dependency Injection, Auto Scanning, Introduction to Spring Boot	• Understand the features and modules of Spring Framework. • Apply Inversion of Control to achieve Dependency Injection using Spring Framework. • Develop Spring application using Java based configuration	9

			Understand Spring Boot and its advantages	
5	Spring 5 Basics with Spring Boot	Abstract class, Abstract method, need and instantiating, Quiz and exercise	• Develop using Autowiring and Scope of the bean, SpringBoot Logging, AOP, Spring Profiles	9

TABLE 3: OVERALL COURSE LEARNING OUTCOME ASSESSMENT CRITERIA AND USE CASES			
LEARNING OUTCOME	**ASSESSMENT CRITERIA**	**Performance Criteria**	**USE CASES**
Implement Java SE 8 features effectively	Evaluate Java SE 8 features and apply them in projects	Demonstrates the ability to use default methods, static methods, lambda expressions, and Streams API for efficient coding.	Develop a program that uses Lambda expressions and Streams API to process and filter large datasets.
Design Maven-based projects	Use Maven for build and dependency management	Creates Maven POM files, integrates plugins, and manages dependencies effectively across lifecycle phases.	Build a multi-module Maven project with plugins for code coverage, testing, and documentation.
Apply Hibernate for Object	Perform CRUD operations	Maps Java classes to database tables using	Create a Hibernate-based

Relational Mapping (ORM)	using Hibernate	annotations and XML files. Executes CRUD operations effectively within an IDE like Eclipse.	library management system with CRUD operations for books, members, and loans.
Develop applications with Spring Framework	Create Spring applications using Dependency Injection	Demonstrates the ability to use Spring IoC for dependency injection and configure beans using XML and annotations.	Build a Spring application for inventory management using dependency injection and REST APIs for CRUD functionality.
Use Spring Boot for rapid application development	Configure and deploy Spring Boot applications	Develops and deploys applications using Spring Boot with appropriate profiles for different environments (e.g., DEV, PROD).	Create a microservice using Spring Boot to handle customer orders, integrating a database and RESTful API endpoints.

TABLE 4: LIST OF FINAL PROJECTS (PROJECTS THAT COMPREHENSIVELY COVER ALL THE LEARNING OUTCOME)	
SL.NO	**FINAL PROJECT**

1	**Problem Statement** Background: This problem statement provides the high-level design of the project that has to be implemented as part of the hands-on assessment in order to complete the course Spring Basics. InfyGo is an airline booking application that provides services to its customers to search for flight details. InfyGo wants a lightweight, loosely coupled application to be implemented using Spring. Let us start with basic implementation using Spring core concepts for the following functionalities • Add Flight • Search Flight As part of the Spring Basics course, let us develop the business tier of this application.

TABLE 5: COURSE ASSESSMENT RUBRICS (TOTAL MARKS: 75)					
ASSESSMENT CRITERIA	**Learning Outcome**	**Fair (1–5)**	**Good (6–10)**	**Excellent (11–15)**	**TOTAL MARKS**
Completion of Java SE 8 programming tasks	Implement Java SE 8 features effectively	Demonstrates limited use of Lambda expressions and APIs	Applies Lambda and Streams with moderate effectiveness	Uses advanced APIs for optimized solutions	15
Build and manage projects using Maven	Design Maven-based projects	Creates basic POM files and manages dependencies partially	Builds projects with moderate plugin and dependency use	Creates optimized builds with comprehensive plugin use	15
Implement ORM with Hibernate	Apply Hibernate for Object Relational	Configures mappings with basic CRUD functionality	Performs mappings and CRUD with moderate	Creates advanced mappings and	15

	Mapping		efficiency	integrates subqueries	
Develop Spring applications	Develop applications with Spring Framework	Creates basic Spring configurations	Develops functional applications using IoC patterns	Builds advanced, scalable applications with profiles	15
Deploy applications with Spring Boot	Use Spring Boot for rapid application developmen t	Deploys basic Spring Boot applications	Configures profiles for moderate deployment scenarios	Implements microservic es with advanced profiles	15