COURSE TEMPLATE

1. Department:	**Department of Computer Science and Engineering**		
Course Name: Middleware Frameworks and ORM	**2. Course Code**	**3. L-T-P**	**4. Credits**
	CSL274	2-0-4	4
5. Type of Course (Check one):	Programme Core ☐	Programme Elective ✔	Open Elective ☐
6. Pre-requisite(s), if any: Enterprise Web Applications			
7. Frequency of offering (check one): Odd ☐	**Even** ✔	Either semester ☐	Every semester ☐

8. Brief Syllabus:

This course covers the programming topics that help design modular and scalable java applications based on REST services using Spring Framework. This course will help in Identifying contrast between traditional and modern ways of application development, thereby recognizing their applications and develop the understanding of core architecture of Spring Framework. By the end of this course the students will have sound knowledge of the concepts that will enable them to simplify development and reduce code complexity with Spring while designing and developing java based interactive modular web applications.

Total lecture, Tutorial and Practical Hours for this course (Take 15 teaching weeks per semester): 90		
	Practice	
Lectures: 30 hours	**Tutorials:** 0 hours	**Lab Work:** 60 hours

9. Course Outcomes (COs)

Possible usefulness of this course after its completion i.e. how this course will be practically useful to him once it is completed.

CO 1	Identifying contrast between traditional and modern ways of application development, thereby recognizing their applications and develop the understanding of core architecture of Spring Framework.
CO 2	Understand and Apply the Aspect Oriented programming Methodology in Spring Framework.
CO 3	Implementation of third-party integration with Spring for the web Application Development.
CO 4	Understand and apply the concepts the basics of Object Relational Mapping in Spring Framework.
CO 5	Understand the Server Side concepts for the development of java based interactive modular web application using Spring Framework.

10. UNIT WISE DETAILS **No. of Units: 5**

Unit Number: 1	**Title:** Spring Core	**No. of hours: 15**

Content Summary:
What is Spring, Brief History of Spring, Why Spring, Before Spring (Loosely Typed Application Example), Basic Spring Example, what is IOC and DI, Spring vs J2EE, Understanding Spring Framework in Detail, Understanding Various Factories in Spring. Spring Setup with JARS only, Spring Setup with Maven, What is Maven, Why we need it, About POM, About Dependency, About Goals, About Plugins, Spring XML Based Approach, Constructor, Injection, Setter Injection, Object Injection, Collection Injection, Bean Inheritance, Bean Life Cycle, Scopes: Singleton and Prototype, Application Context Aware, Code By Interface, Auto Wire and Its Types, Using Property File, Spring Annotation Based Approach, Stereotype, Auto Wire and it types and Qualifiers, Default Bean Name, Qualifiers with Constructors, Using Property file, Constructor Injection, Setter Injection, Object Injection, Collection Injection, Bean Inheritance, Bean Life Cycle, Scopes: Singleton and Prototype, Spring Java Config Approach: @Bean, @Configurable, @Primary, Using Property file: Constructor Injection, Setter Injection, Object Injection, Collection Injection, Bean Inheritance, Bean Life Cycle, Scopes, Singleton and Prototype.

Unit Number: 2	**Title: Aspect Oriented programming**	**No. of hours: 3**

Content Summary:
Overview, Advice Types, Point cuts, Ordering Aspects, Join Point

Unit Number: 3	**Title: Spring MVC**	**No. of hours: 5**

Content Summary:
Spring integration with JDBC, Spring integration with Logger, Spring Integration with Mail, Spring integration with JPA.

Unit Number: 4	**Title**: **Spring Integration**	**No. of hours: 5**

Content Summary:
Spring integration with Logger, Spring Integration with Mail, Spring integration with JPA

Unit Number: 5	**Title:** Hibernate Basics	**No. of hours: 2**

Content Summary:
ORM Basics, Hibernate vs JDBC, Setup with Maven, CRUD Operations.

11. Brief Description of Self-learning components by students (through books/resource material etc.):

Spring Testing and Spring Web Services

12. Books Recommended:

Textbooks:

Iuliana Cosmina, Rob Harrop, Chris Schaefer, Clarence Ho, “Pro Spring 5”, Apress Publisher, 5th Edition, 2017.

Reference Books:

Craig Walls, “Spring in Action”, Manning, 5th Edition, 2018

Reference Websites: (nptel, swayam, coursera, edx, udemy, lms, official documentation weblink)

- https://maven.apache.org/guides/getting-started/
- https://spring.io/

- https://www.coursera.org/learn/web-development-with-java-spring-framework
- https://www.udemy.com/course/spring-framework-5-beginner-to-guru/

Experiential Learning Component

S No.	Topic	Type of Submission/Assessment Mode	Cos Covered
1	Real time case studies on implementation of Injections, Singelton and Prototype scopes	Evaluation in Tutorial Class through a one or two page write up	**CO1**
2	Case Study to analyze the statement: As the number of aspects in a feature grows, there is a noticeable decrease in code readability and maintainability, Also document where AspectJ is unsuitable for implementing features of refactored legacy applications and explain why.	Evaluation in Tutorial class with practical assessment through group presentation	**CO2**
3	Guest lecture by Industrial Expert "Spring Framework: a recent demand of market"	Attended Certificate submission	**CO1-CO5**
4	Online Quiz	Online Submission	**CO1-CO5**
5	Mini Project Submission	End- term project submission evaluation	**CO1-CO5**

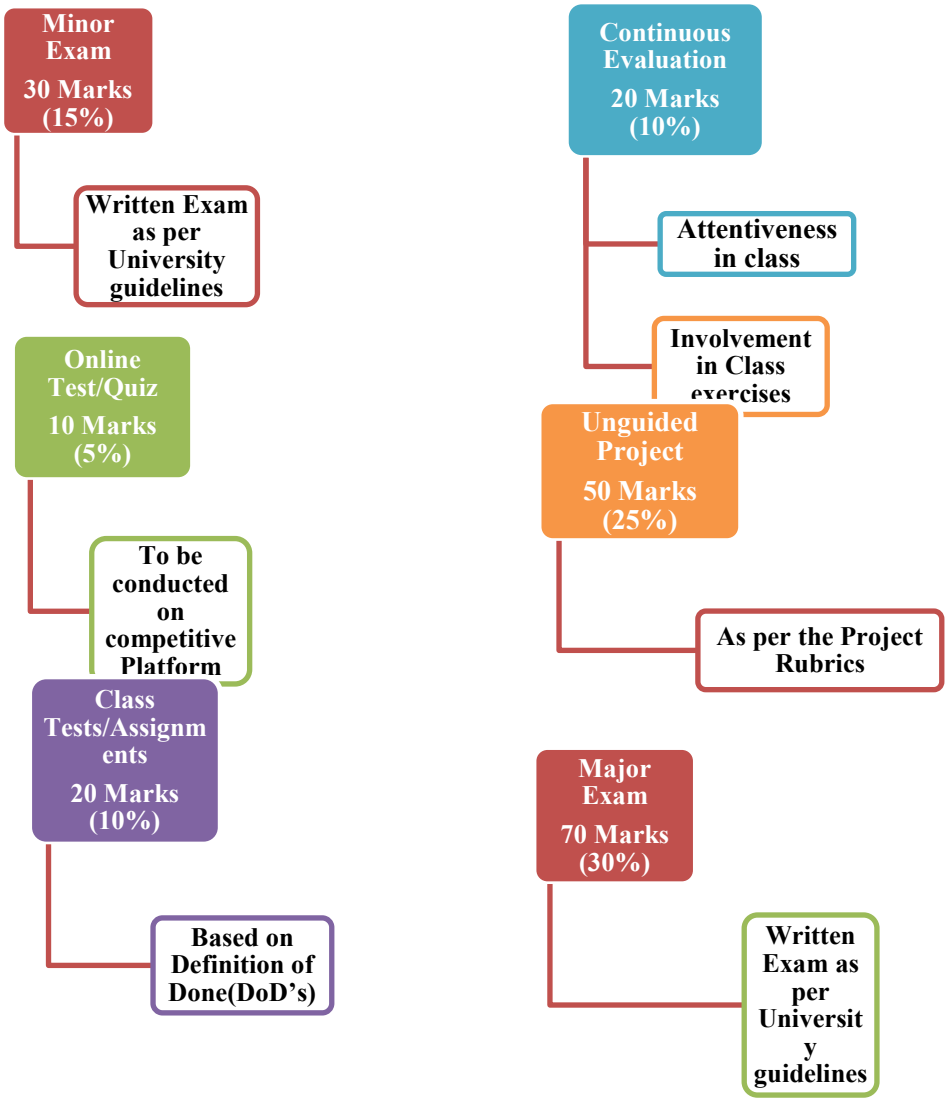
Practical Content

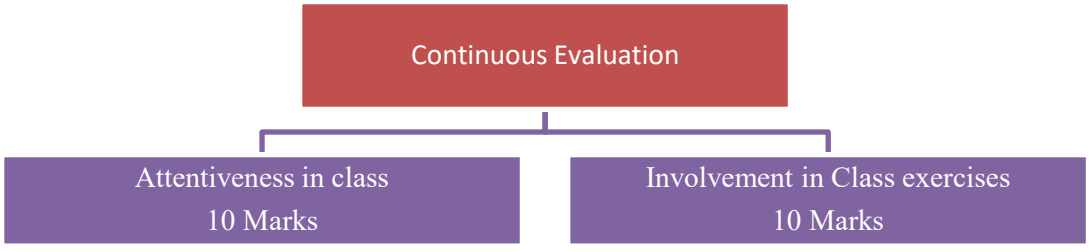
Sr. No.	Title of the Experiment	Software/ Hardware based	Unit covered	Time Required
1.	Loosely coupled and tightly coupled	Software	1	2 hours
2.	DB Factory	Software	1	2 hours
3.	LOS(Loan Origination System)	Software	1	2 hours
4.	Create Singleton Bean	Software	1	2 hours
5.	Effect of singleton and prototype	Software	2	2 hours
6.	Create a Loan management system	Software	2	5 hours
7.	If first bean is singleton and another is prototype and vice-versa what will be happen	Software	3	5 hours
8.	Implementation of Injections: Constructor, Setter	Software	3	5 hours
9.	Create a Maven Project	Software	4	5 hours
10.	Implementation of auto wiring: ByType, ByName Create two classes A &B and do the DI of A with B using By Type. Create another class C and do its DI with A using By Name. Create one more class i.e. Caller class which is main class of the project	Software	4	5 hours
11.	Collection Injection:LIST,SET,MAP	Software	5	5 hours
12.	Implement Loosely Coupled app using Annotation.	Software	5	5 hours
Value Added Experiments				
1.	Think how to inject the static members using spring xml approach and also try to find out the way by which we can	Software	5	5 hours

	make the object of Scanner class without using new keyword using xml only.			
2.	Write a program to make a Loan class and the condition is that it's object will be formed only once and every time when user need object of Loan class then it is referencing to the same instance of Loan class. Hint :- Singleton	Software	5	5 hours
3.	Make a loosely coupled JAVA program for making DAO (Data Access Object) POJO Class. For this make an Interface which contains the CRUD (Create/Read/Update/Delete) methods. And then make three classes which implements the interface for CRUD with OracleDAO class, MySQLDAO class and the PosGresDAO class. Consume DAO in Loosley Coupled Manner.	Software	5	5 hours

Project (To be done as individual/in group): Yes (List of projects with their Definition of Done (DoDs) will be provided by faculty and will be evaluated by external faculty teaching the same course in the department)

Evaluation Scheme





Peer Mentoring	**5**
Discipline	**5**

Shown 90 - 100 % of the exercises	**14-15**
Shown 80 - 90 %	**12-13**

Participation in class discussions	**5**
Total	**15**

Shown 70% - 80 %	**9-11**
Shown 60 - 75 %	**6-8**
Shown <60%	**5**

Component	**Max Marks (20)**					**Total Marks**
	DoD 1	**DoD 2**	**DoD 3**	**DoD 4**	**DoD 5**	
Assignment 1	**3**	**3**	**5**	**5**	**4**	**20**
Online course	**Online Test Completion**					**20**
Competitive Programming	**Badges / Stars Earned**					**20**

Evaluation Scheme

S. No.	TYPE OF COURSE	PARTICULAR	ALLOTTED RANGE OF MARKS	PASS CRITERIA
1	Project Based Course (L-T-P/L-T-0/L-0-P/L-0-0)	Minor Test	15%	Must Secure 30% Marks Out of Combined Marks of End Term Project Plus Major Test & Minor Test with Overall 40% Marks in Total.
		Major Test	35%	
		Unguided Project	25%	
		Continuous Evaluation	10%	
		Class Test/ Assignment	10%	
		Online Test/Quiz	5%	

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	3	3	3	5	3	3	4	4	3	2	2	4	1	3
CO2	1	2	2	2	4	2	2	2	3	3	3	3	3	1	3
CO3	3	3	3	4	5	4	4	4	4	4	4	4	3	1	3
CO4	2	3	2	2	3	2	2	4	4	3	3	3	5	1	3
CO5	2	3	5	2	4	4	4	4	4	3	4	3	3	1	3

Mapping of PO's and CO's

List of Case Studies

1. Think how to inject the static members using spring xml approach and also try to find out the way by which we can make the object of Scanner class without using new keyword using xml only.
2. When working with enterprise applications, you often want to refer to modules of the application and particular sets of operations from within several aspects. Design a suitable solution to solve this problem.
3. Create a spring project implementing Restful API –
    - The program shall contain a controller class with proper annotation along with the implementation class.
    - There must be proper Database connection.
    i. To validate the logged in user.
        - User shall provide username and password in the input json.
        - These username and password shall be retrieved in Controller class and passed onto implementation class where it shall be validated with the details present in database.
        - On successful validation, the output shall be JSON format with value as valid user else Invalid user. along with role of the user

    ii. Program shall get all the details of the users stored in the database along with their role.
4. If I want to design my software application that is adaptable to various languages then what are the changes do I need to incorporate into my application, write all the changes that are needed with an example.