

	<b>ITER, SIKSHA 'O' ANUSANDHAN</b> <b>Deemed to be University</b>		<b>LESSON PLAN</b>
Programme	<b>B.Tech.</b>	Academic Year	<b>2023-24</b>
Department	<b>CSE/CSIT</b>	Semester	<b>4<sup>TH</sup></b>
Instructor	<b>Rasmiranjan Mohakud</b>	Grading Pattern	<b>5</b>
Subject Code	<b>CSE 3141</b>		
Subject Name	<b>Computer Science Workshop 2</b>		

### Text Books(s):

- (1) Full Stack Java Development with Spring MVC, Hibernate, jQuery, and Bootstrap by Mayur Ramgir, Wiley India.

**Course Format:** 8 Lab Contact Hours / week

<b>Course Outcomes</b>	Students will be able to	
	<b>CO1</b>	Understanding Object-oriented programming, generic and collection class and applying it to solve different problems.
	<b>CO2</b>	Understanding Error handling, garbage collection, string, I/O operation, and file management of java. And apply it to solve related problems.
	<b>CO3</b>	Learning different data structure and applying it to solve different problems and analyze its effectiveness on different problem-solving. Understanding and applying Lambdas and Functional programming using Java.
	<b>CO4</b>	Understanding multithreading and reactive programming of java, and applying it to solve related problems.
	<b>CO5</b>	Learning spring and spring MVC of Java and applying it to solve different problems.
	<b>CO6</b>	Understanding and learning Hibernate and applying it to solve different problems.

Lab	Lessons/Topics to be covered	Book reference (sections)	Mapping with COs
<b>Week #1:</b>			
<b>Lab#1</b>	Introduction to course and POs and COs. Introduction to Object-Oriented Programming	1.1-1.3 (pg.1-3)	All COs
<b>Lab#2</b>	Object-Oriented Programming Principles, Encapsulation, Abstraction, Inheritance, Polymorphism. Explain it with suitable examples. Solve some problems related to these concepts.	1.4 (pg.3-8)	<b>CO1</b>

<b>Lab#3</b>	Object-Oriented Programming Principles in Application, Understanding an Interface, Overriding and Overloading. Explain it with suitable examples. Solve some problems related to these concepts.	1.5-1.6 (pg.8-11)	<b>CO1</b>
<b>Week #2:</b>			
<b>Lab#4</b>	Coupling and Cohesion. Explain it with suitable examples. Solve some problems related to these concepts.	1.7-1.10 (pg.12-18)	<b>CO1</b>
<b>Lab#5</b>	Introduction to Generics and Collections. Generic Programming, Benefits of Generics, Using Generics in Java, Generic Methods, Overriding toString(), hashCode(), and equals().	2.1-2.2 (pg.19-20)	<b>CO1</b>
<b>Lab#6</b>	Collections in Java, Benefits of Java Collections, Collection Interfaces, Collections Classes.	2.3-2.5 (pg.20-22)	
<b>Week #3:</b>			
<b>Lab#7</b>	Implementing Collection Classes, Map Interface. Explain it with suitable examples. Solve some problems related to these concepts.	2.6-2.7 (pg.22-30)	<b>CO1</b>
<b>Lab#8</b>	Set Interface. Explain it with suitable examples. Solve some problems related to these concepts.	2.8-2.9 (pg.31-35)	<b>CO1</b>
<b>Lab#9</b>	List Interface. Explain it with suitable examples. Solve some problems related to these concepts.	3.1-3.3 (pg.37-49)	<b>CO1</b>
<b>Week #4:</b>			
<b>Lab#10</b>	Queue Interface, Stream API. Explain it with suitable examples. Solve some problems related to these concepts.	3.4-3.5 (pg.50-56)	<b>CO1</b>
<b>Lab#11</b>	List of Key Methods for Arrays and Collections, Arrays (java.util.Arrays), Collections (java.util.Collections). Explain it with suitable examples. Solve some problems related to these concepts.	3.6-3.8 (pg.57-61)	<b>CO1</b>
<b>Lab#12</b>	Key Methods for List, Set, Map, and Queue. Solve some problems related to these concepts.	3.9-3.10 (pg.62-65)	<b>CO1</b>
<b>Week #5:</b>			
<b>Lab#13</b>	Error Handling, Introduction, Understanding Error Handling, Logical Errors. Explain it with suitable examples. Solve some problems related to these concepts.	3.11 (pg.66-70)	<b>CO2</b>
<b>Lab#14</b>	Syntactical Errors, Semantic Errors. Explain it with suitable examples. Solve some problems related to these concepts.	4.1-4.4 (pg.73-74)	<b>CO2</b>
<b>Lab#15</b>	Importance of Error Handling, Checked versus Runtime Exceptions. Explain it with suitable examples. Solve some problems related to these concepts.	4.5-4.8 (pg.74-82)	<b>CO2</b>

<b>Week #6:</b>			
<b>Lab#16</b>	Garbage Collection, Introduction, Garbage Collection in Java, Major Garbage Collection, G1 and CMS Garbage Collectors, Advantages of Garbage Collection in Java.	4.9-4.13 (pg.85-94)	<b>CO2</b>
<b>Lab#17</b>	Making Objects Eligible for Garbage Collection, JEP 318 – Epsilon: A No-Op Garbage Collector.	4.14-4.17 (pg.95-102)	<b>CO2</b>
<b>Lab#18</b>	Strings, I/O Operations, and File Management, Introduction, Role of Strings in Java, Types of String Operations. Explain it with suitable examples. Solve some problems related to these concepts.	4.18-4.21 (pg.104-114)	<b>CO2</b>
<b>Week #7:</b>			
<b>Lab#19</b>	StringBuilder and StringBuffer. Explain it with suitable examples. Solve some problems related to these concepts.	5.1-5.3 (pg.117-118)	<b>CO2</b>
<b>Lab#20</b>	Java I/O. Explain it with suitable examples. Solve some problems related to these concepts.	5.4-5.5 (pg.119-122)	<b>CO2</b>
<b>Lab#21</b>	Data Structure and Integration in Program, Introduction, Introduction to Data Structures, Classification of Data Structures.	5.6-5.7 (pg.126-149)	<b>CO3</b>
<b>Week #8:</b>			
<b>Lab#22</b>	Implement graph data structure and its traversing.	5.8-5.9 (pg.153-171)	<b>CO3</b>
<b>Lab#23</b>	Lambdas and Functional Programming, Introduction, Fundamental Concepts of Functional Programming. Functional Programming in Java. Explain it with suitable examples. Solve some problems related to these concepts.	5.9 (pg.172-175)	<b>CO3</b>
<b>Lab#24</b>	Lambdas, Explain it with suitable examples. Solve some problems related to these concepts.	6.1-6.2 (pg.177-179)	<b>CO3</b>
<b>Week #9:</b>			
<b>Lab#25</b>	Lambdas, Explain it with suitable examples. Solve some problems related to these concepts.	6.3-6.6 (pg.179-187)	<b>CO3</b>

<b>Lab#26</b>	Date and Time API Explain it with suitable examples. Solve some problems related to these concepts.	6.7-6.10 (pg.190-200)	<b>CO3</b>
<b>Lab#27</b>	Multithreading and Reactive Programming, Introduction, Reactive Programming, Multithreading in Java, Programming with Multithreading.	6.11-6.13 (pg.201-208)	<b>CO4</b>
<b>Week #10:</b>			
<b>Lab#28</b>	Concurrency . Explain it with suitable examples. Solve some problems related to these concepts.	6.14-6.17 (pg.209-219)	<b>CO4</b>
<b>Lab#29</b>	Concurrency. Explain it with suitable examples. Solve some problems related to these concepts.	6.18 (pg.223-229)	<b>CO4</b>
<b>Lab#30</b>	Understanding Deadlock. Explain it with suitable examples. Solve some problems related to these concepts.	7.1-7.4 (pg.231-239)	<b>CO4</b>
<b>Week #11:</b>			
<b>Lab#31</b>	Concurrent Data Structures, Multithreading Examples, Designing Concurrent Java Programs.	8.1-8.2 (pg.243-248)	<b>CO4</b>
<b>Lab#32</b>	Introduction to Spring and Spring MVC, Spring Framework, Spring Architecture, Spring MVC.	9.1 (pg.251-257)	<b>CO5</b>
<b>Lab#33</b>	Interception, Chain of Resolvers, View Resolution, Multiple View Pages, Multiple Controllers.	10.1-10.4 (pg.259-278)	<b>CO5</b>
<b>Week #12:</b>			
<b>Lab#34</b>	Model Interface, RequestParam, Form Tag Library, Form Text Field	10.5-10.7 (pg.279-292)	<b>CO5</b>
<b>Lab#35</b>	CRUD Example, File Upload in Spring MVC, Validation in Spring MVC, Validation with Regular Expression, Validation with Numbers.	11.1-11.2 (pg.299-303)	<b>CO5</b>
<b>Lab#36</b>	Introduction to Hibernate, Architecture, Installation and Configuration	11.3-11.5 (pg.304-307)	<b>CO6</b>

**Week #13:**

<b>Lab#37</b>	Java Objects in Hibernate, Inheritance Mapping, Collection Mapping, Mapping with Map, Hibernate Query Language,	11.6-11.8 (pg.309-326)	<b>CO6</b>
<b>Lab#38</b>	Caching, Spring Integration	11.9-11.11 (pg.327-335)	<b>CO6</b>

☞ Few Groups will be assigned to conduct Module based Experiments (Verification / Understanding domain knowledge) and Some groups need to be assigned with design projects (Analysis/ Implementation).

**☆ List of Projects:****1. E-Commerce Website**

Description: Develop a basic e-commerce website that showcases products, allows users to browse and search for items, add them to a cart, and proceed to checkout. Implement both front-end and back-end functionalities. Apply HTML, CSS, Bootstrap, Java for back-end processing, and Hibernate for managing product data in a database.

**2. Personal Portfolio Website**

Description: Create a personal portfolio website for showcasing your skills and projects. Design an interactive and responsive user interface using HTML, CSS, and Bootstrap. Implement a contact form that sends messages to an email address using Java-based back-end processing.

**3. Online Quiz Application**

Description: Build an online quiz application where users can select a quiz topic, answer multiple-choice questions, and receive instant feedback on their performance. Design an engaging user interface using HTML, CSS, and Bootstrap. Implement the quiz logic using Java for back-end processing and store quiz data using Hibernate.

**4. Task Manager Web App**

Description: Develop a task manager web application that allows users to create, update, and delete tasks. Implement user authentication and authorization for accessing tasks. Use HTML, CSS, Bootstrap for the front-end, and Java with Hibernate for the back-end to manage task data in a database.

**5. Blogging Platform**

Description: Create a blogging platform where users can write and publish articles. Design an intuitive front-end using HTML, CSS, and Bootstrap. Develop a user authentication system and allow registered users to create, edit, and delete their posts. Use Java and Hibernate to manage user and article data.

**6. Weather Dashboard**

Description: Build a weather dashboard that allows users to search for the weather forecast of different cities. Utilize APIs to fetch weather data and display it in a user-friendly interface. Design the dashboard using HTML, CSS, and Bootstrap. Implement back-end functionality in Java to handle API requests and responses.

These project ideas align with the course outcomes have provided and offer opportunities for students to apply their knowledge of full-stack web development, Java programming, Hibernate, and other relevant concepts. Students can work on these projects to gain practical experience and demonstrate their skills.