Semester: 7th			
Paper code: OAE417P	L	T/P	Credits
Subject: Advanced Java Programming Lab	0	2	1
Marking Scheme:			
1. Teachers Continuous Evaluation: As per university examination norms from time to time			
2. End term Examination: As per university examination norms from time to time			
INSTRUCTIONS TO EVALUATORS: Maximum Marks: As per university norms			
1. This is the practical component of the corresponding theory paper.			
2. The practical list shall be notified by the teacher in the first week of the class commencement under the intimation to the office of the HOD/ Institution in which they appear is being offered from the list of practicals below.			
3. Instructors can add any other additional experiments over and above the mentioned in the experiment list which they think is important.			
4. At least 8 experiments must be performed by the students.			
Course Objectives:			
1.	Develop a deep understanding of advanced Java concepts, such as multi-threading, networking, and database connectivity, to build robust and efficient applications		
2.	Gain practical experience by working on real-world Java projects, which involve solving complex problems and implementing solutions using advanced Java features.		
Course Outcomes:			
CO1	Achieve proficiency in utilizing advanced Java features, including multithreading, socket programming, JDBC, and JavaFX, to develop high-performance applications.		
CO2	Be capable of designing and building robust, scalable, and secure applications that leverage advanced Java programming techniques for real-world use cases.		

Course Outcomes (CO) to Programme Outcomes (PO) Mapping

(Scale 1: Low, 2: Medium, 3: High)

CO/PO	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12
CO1	2	2	2	3	3	-	1	-	1	-	-	2
CO2	2	2	-	3	3	-	-	-	-	-	1	1

List of Experiments:

1. Write a java program of thread synchronization, inter-thread communication, and thread pooling.
2. Implement a client-server application using Java's networking APIs.
3. Design a calculator, a simple text editor, or a graphical game with user interaction and visual components. Explore event handling, layout managers, and UI design principles.
4. Implement functionalities like data retrieval, insertion, deletion, and updating records. Explore concepts like JDBC, SQL queries, and database transactions.
5. Utilize third-party libraries or frameworks in Java programming. Choose a popular library (e.g., Apache Commons, Gson, Log4j) and develop programs that showcase its features and functionality.
6. Write a java program to writes objects to a file in a serialized format and then reads and reconstructs the objects from the file.

7. Write a java program that uses reflection to inspect and modify the behavior of objects based on user input or external configuration.

8. Implement generic methods to perform operations like sorting, searching, or filtering on generic collections.

9. Design custom annotations and use them in a Java program to provide additional metadata and define behavior.

10. Write a java program to Integrate Java with native code by using the JNI (with native libraries written in C/C++).

11. Implement functional programming concepts and solve problems related to data manipulation, filtering, or mapping.