Practice Assignment

Module 4	
Sr. No.	**Questions**
1	Imagine you are designing a real-time online multiplayer game using Java. How would you implement multithreading and concurrency to ensure smooth gameplay and prevent performance bottlenecks? Discuss potential challenges and solutions.
2	Generics play a crucial role in enhancing type safety and code reusability in Java. Describe a scenario where you would use generics to design a flexible and robust data structure. Provide code examples to illustrate your point.
3	Annotations are widely used in Java to add metadata and instructions to the code. Explain how you would create a custom annotation that can be applied to methods to track their execution time. Provide an example of how you would use this annotation in a class.
4	Reflective programming allows Java programs to inspect and manipulate their own structure during runtime. Discuss a practical use case where you would utilise reflection to dynamically load and execute classes, providing a step-by-step explanation of the process.
5	You are developing a web application that requires interaction with a relational database. Describe how you

	would use Java Database Connectivity (JDBC) to establish a connection to the database, perform CRUD operations, and handle exceptions effectively.
6	In a complex financial software system, explain how you would implement a thread-safe singleton design pattern using Java's multithreading features. Discuss the benefits of using this pattern and any potential pitfalls to be aware of.
7	Consider a scenario where you need to process a large dataset stored in a CSV file and save the results in a database. Outline the steps you would take to achieve this using file I/O operations, multithreading, and JDBC. Highlight the challenges and optimisations involved.
8	Describe an application scenario where using generics can lead to a significant reduction in code duplication and enhancement of type safety. Provide examples of classes and methods you would create to demonstrate the implementation of generics in this context.
9	Annotations can be employed for both compile-time and runtime processing. Explain how you would create a custom annotation that triggers specific actions during runtime, and discuss the potential use cases and benefits of such an annotation in a Java application.
10	You are building a Java application that needs to dynamically load classes and invoke their methods based on user input. Elaborate on how reflection can be employed to achieve this functionality, discussing both the advantages and potential security concerns.