Course Design Document

Course Code	
Course Name	Logging, Refactoring and Best Practices

Duration (in days)	1	Proficiency Level	Fundamental
Pre-requisites	• Core Java	Target Audience	Campus Hires

Learning Outcome

At the end of the program, participants will learn

- Refactoring API Code
- Different Java logging frameworks
- Working with loggers in a Java EE application

Day wise Session Plan

Day	Unit	Objectives	Hours
1	Refactoring API Code	• Discover the key concepts covered in this course • Describe the need for refactoring code and some best practices for refactoring • Describe the benefits of refactoring code • Describe where code technical debt comes from and how to manage it • Describe when code refactoring makes sense and when it does not • Describe how refactoring relates to test-driven development • Describe available refactoring tools to assist in the refactoring process • Classify the various code refactoring methods and features • Describe the code refactoring methods of coordinating data • Describe the code refactoring methods of simplifying Boolean expressions • Describe the code refactoring methods of simplifying method calls • Describe the code refactoring methods for simplifying class hierarchies • Describe the process of refactoring code to move towards a microservice architecture	4
2	Packaging Applications & Logging	• Describe Java logging and logging frameworks • Describe how to use the java.util.logging.Logger class in Java EE applications • Describe logging handlers in Java EE applications	4

		• Referencing the log service, describe how logging is configured and logs viewed for Java EE applications • Create a Logger instance and add it to a Java EE application • Practice creating and working with loggers in a Java EE application	