**Northeastern University – Silicon Valley**

CS 6650 Scalable Dist Systems

**Homework #4** [100 points]

***INSTRUCTIONS: Please provide clear explanations in your own sentences, directly answering the question, demonstrating your understanding of the question and its solution, in depth, with sufficient detail. Submit your solutions [PDF preferred]. Include your full name. Do not email the solutions.***

Answer the following questions using explanation and diagrams as needed. No implementation needed.

Study **Chapter 16 from** Coulouris Book

1. 16.1 [10 points]
2. 16.2 [10 points]
3. Concurrency Control and Java Apps
4. Explain the 3 different concurrency control methods (Ch 16 Coulouris Book – Locks based, Optimistic and Time-stamp Ordering) brifely. Compare and contrast how they achieve concurrency control. [10 points]
5. **Java Concurrency In Practice** by Brian Goetz is the practice manual for Dist Systems. Study Chapter 6 (Task Execution) which is a small chapter.

Identify 5 different programming devices (e. g. Tasks, Exec Framework) used in Java for Task Execution. Then, briefly explain how you might use these to implement a basic

Locks based concurrency control in a simple Web App. [15 points]

1. Refer to Oracle Database Administrator’s Guide page 29-23 [15 points]

**Transaction Processing in a Distributed System**

<https://docs.oracle.com/html/E25494_01/ds_concepts004.htm> But get the PDF, it is easier to read.

What are distributed transactions in Oracle DB? How are the different fro Remote Transactions? Give examples. How does Oracle DB use Naming service and 2-Phase Atomic COmit Protocols to manage distributed transactions?

1. Study Alibaba Fescar note. How does Fescar manage distributed transactions? What protocols are used? Describe in detail. How does it avoid Deadlock? [20 points]
2. Transaction API (JTA) allows applications to perform distributed transactions. For a small e-commerce retail store, explain a design for implementation using JTA. Show the key Classes and how you use them to resolve concurrency issues and deadlock. [20 points]