**PART 2 ‐ Practical Component 2 ‐ 30%**

The same group will extend the already developed project for the following features. Each group is required to submit on Brightspace the solution for the practical task. The submission must include both the client and the server NetBeans projects to allow for compilation and testing.

**Description**

Suppose that multiple clients can now schedule the timetable at the same time. Provide all the functionalities in Part 1 by adding the multi-processor architecture support using the Multithreading concept.

To implement this, the server application must start a new Thread for each connection established with a client.

**Each thread on the server side will then:**

1. Receives the requests/messages sent by the client application.

2. Calls the appropriate method to perform the action indicated in the message.

3. Replies to the client with a piece of information according to the action performed.

Your applications must ensure synchronization and control access to the memory‐based data collection (e.g., ArrayList, HashMap object) managed on the server side. This time your server will also be a GUI application.

In addition to this, implement a new feature in your applications where clients can request ‘early lectures’ and the server brings the classes towards morning timings.

For example,

if two classes are scheduled on Tuesday between 13:00 –15:00 and 15:00 – 17:00 and three classes are scheduled on Wednesday between 11:00 – 13:00, 13:00 –15:00, and 15:00 – 17:00, then an ‘early lectures’ request from client will cause the server to shift those classes in early mornings (e.g., between 9:00 and 13:00 for Tuesday classes and between 9:00 and 15:00 for Wednesday classes) ONLY if the early morning timeslots are not occupied. You are expected to implement this functionality using a divide-and-conquer algorithm, where a different thread is shifting the classes to early mornings for each day (i.e., different threads created to deal with separate days in a week).