Experiment 4

```
package dc.berkeley_algo;
import java.util.Date;
import java.text.ParseException;
import java.text.SimpleDateFormat;
public class berkeley {
   public static void berkeleyAlgo(String servertime, String time1, String time2) {
      System.out.println("Server Clock   = " + servertime);
      System.out.println("Client Clock 1 = " + time1);
      System.out.println("Client Clock 2 = " + time2);
      SimpleDateFormat sdf = new SimpleDateFormat("mm:ss");
      try {
         /* Converting time to Milliseconds */
         long s = sdf.parse(servertime).getTime();
         long t1 = sdf.parse(time1).getTime();
         long t2 = sdf.parse(time2).getTime();
         /* Calcuating time differences w.r.t server */
         long st1 = t1 - s;
         System.out.println("t1 - s = "+st1/1000);
         long st2 = t2 - s;
         System.out.println("t2 - s = "+st2/1000);
         /* Fault tolerant Average */
         long aveg = (st1 + st2 + 0) / 3;
         System.out.println("(st1 + st2 + 0)/3 = "+aveg/1000);
         /* Adjustment */
         long adjserver = aveg+s;
         long adj_t1 = aveg-st1;
         long adj_t2 = aveg-st2;
         System.out.println("t1 adjustment = "+adj_t1/1000);
         System.out.println("t2 adjustment = "+adj_t2/1000);
         /* sync clock */
         System.out.println("Synchronized Server Clock  = "+sdf.format(new Date(adjserver)));
         System.out.println("Synchronized Client1 Clock = "+sdf.format(new Date(t1+adj_t1)));
         System.out.println("Synchronized Client2 Clock = "+sdf.format(new Date(t2+adj_t2)));
      } catch (ParseException e) {
         e.printStackTrace();
      }
   }
   public static void main(String[] args) {
      berkeleyAlgo("03:00", "03:25", "02:50");
   }
}
```

Output

```
Server Clock   = 03:00
Client Clock 1 = 03:25
Client Clock 2 = 02:50
t1 - s = 25
t2 - s = -10
(st1 + st2 + 0)/3 = 5
t1 adjustment = -20
t2 adjustment = 15
Synchronized Server Clock  = 03:05
Synchronized Client1 Clock = 03:05
Synchronized Client2 Clock = 03:05
```