

Department of Electrical and Computer Engineering CSE215: Programming Language II

Assignment 02

Name : Md. Misbah Khan

Student ID : 2132089642

Course Title : Programming Language II

Course Code : CSE215

Section : 13

Faculty : Dr. Mohammad Abu Yousuf

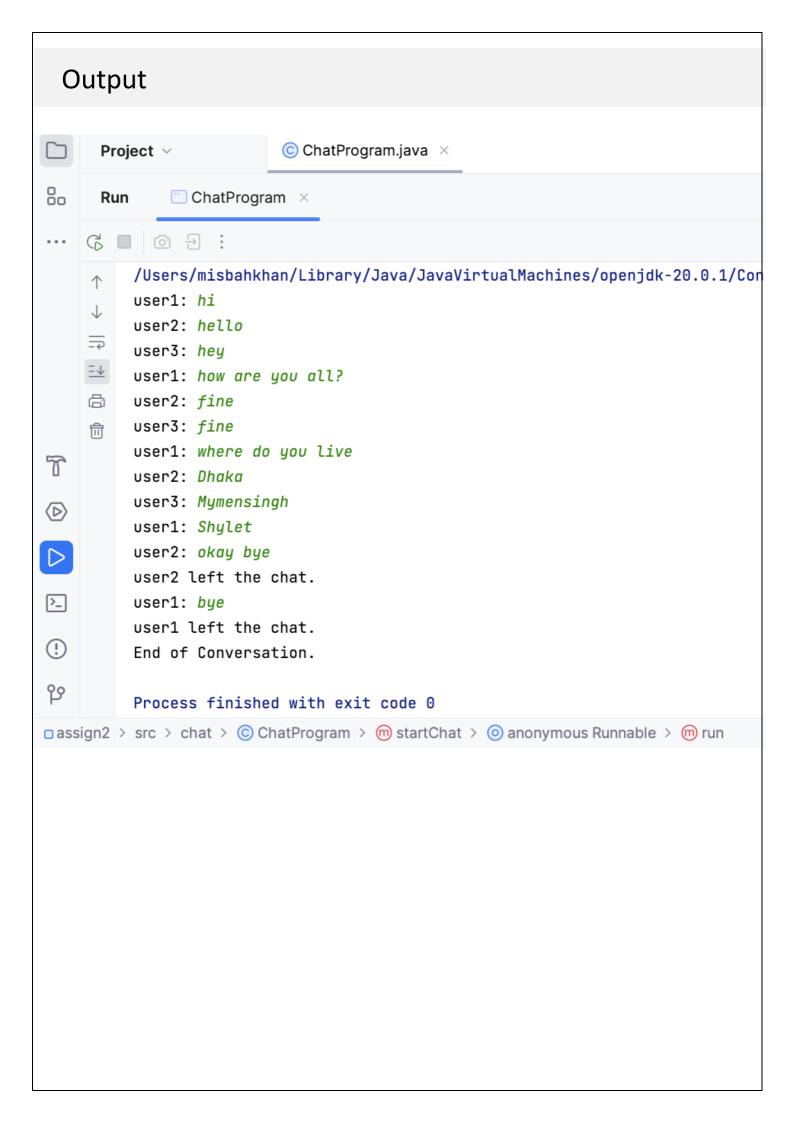
Submission Date: 04/06/2023

Assignment Topic: Design and Implement of a Chat System using Synchronized

Multithreaded Programming

Code

```
package chat;
import java.util.*;
public class ChatProgram {
    private static Object lock = new Object();
    private static List<String> activeUsers = new ArrayList<>();
    public static void users(){
        activeUsers.add("user1");
        activeUsers.add("user2");
        activeUsers.add("user3");
    public static void main(String[] args) {
        users();
        startChat();
    private static int writer = 0;
    private static void startChat() {
        for (int i = 0; i < activeUsers.size(); i++) {</pre>
            String USER = activeUsers.get(i);
            Scanner sc = new Scanner(System.in);
            Thread thread = new Thread(new Runnable() {
                @Override
                public void run() {
                    while (true) {
                         synchronized (lock) {
                             while (!USER.equals(activeUsers.get(writer))) {
                                 try {
                                     lock.wait();
                                 } catch (InterruptedException e) {
                                     System.out.println(e);
                             System.out.print(USER + ": ");
                             String message = sc.nextLine();
                             if (message.indexOf("bye") != -1) {
                                 activeUsers.remove(USER);
                                 System.out.println(USER + " left the chat.");
                                 if (activeUsers.size() <= 1) {</pre>
                                     System.out.println("End of Conversation.");
                                     stopChat();
                                 }
                             int bye = -1;
                             if (bye == -1) {
                                 writer = (writer + 1) % activeUsers.size();
                             } else {
                                 writer = bye;
                                 bye = -1;
                             lock.notifyAll();
                        }
                    }
                }
            });
            thread.start();
        }
    private static void stopChat(){
        System.exit(0);
      }
}
```



Flow Chart

