***2PC Protocol***

***Server:***

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class Server

{

boolean closed=false,inputFromAll=false;

List<clientThread> t;

List<String> data;

Server()

{

t = new ArrayList<clientThread>();

data= new ArrayList<String>();

}

public static void main(String args[])

{

Socket clientSocket = null;

ServerSocket serverSocket = null;

int port\_number=1521;

Server ser=new Server();

try

{

serverSocket = new ServerSocket(port\_number);

}

catch (IOException e)

{

System.out.println(e);

}

while(!ser.closed)

{

try

{

clientSocket = serverSocket.accept();

clientThread th=new clientThread(ser,clientSocket);

(ser.t).add(th);

System.out.println("\nNow Total clients are : "+(ser.t).size());

(ser.data).add("NOT\_SENT");

th.start();

}

catch (IOException e)

{

}

}

try

{

serverSocket.close();

}

catch(Exception e1)

{

}

}

}

class clientThread extends Thread

{

DataInputStream is = null;

String line;

String destClient="";

String name;

PrintStream os = null;

Socket clientSocket = null;

String clientIdentity;

Server ser;

public clientThread(Server ser,Socket clientSocket)

{

this.clientSocket=clientSocket;

this.ser=ser;

}

public void run()

{

try

{

is = new DataInputStream(clientSocket.getInputStream());

os = new PrintStream(clientSocket.getOutputStream());

os.println("Enter your name.");

name = is.readLine();

clientIdentity=name;

os.println("Welcome "+name+" to this 2 Phase Application.\nYou will receive a vote Request now...");

os.println("VOTE\_REQUEST\nPlease enter COMMIT or ABORT to proceed : ");

for(int i=0; i<(ser.t).size(); i++)

{

if((ser.t).get(i)!=this)

{

((ser.t).get(i)).os.println("---A new user "+name+" entered the Appilcation---");

}

}

while (true)

{

line = is.readLine();

if(line.equalsIgnoreCase("ABORT"))

{

System.out.println("\nFrom '"+clientIdentity+"' : ABORT\n\nSince aborted we will not wait for inputs from other clients.");

System.out.println("\nAborted....");

for(int i=0; i<(ser.t).size(); i++)

{

((ser.t).get(i)).os.println("GLOBAL\_ABORT");

((ser.t).get(i)).os.close();

((ser.t).get(i)).is.close();

}

break;

}

if(line.equalsIgnoreCase("COMMIT"))

{

System.out.println("\nFrom '"+clientIdentity+"' : COMMIT");

if((ser.t).contains(this))

{

(ser.data).set((ser.t).indexOf(this), "COMMIT");

for(int j=0;j<(ser.data).size();j++)

{

if(!(((ser.data).get(j)).equalsIgnoreCase("NOT\_SENT")))

{

ser.inputFromAll=true;

continue;

}

else

{

ser.inputFromAll=false;

System.out.println("\nWaiting for inputs from other clients.");

break;

}

}

if(ser.inputFromAll)

{

System.out.println("\n\nCommited....");

for(int i=0; i<(ser.t).size(); i++)

{

((ser.t).get(i)).os.println("GLOBAL\_COMMIT");

((ser.t).get(i)).os.close();

((ser.t).get(i)).is.close();

}

break;

}

}//if t.contains

}//commit

}//while

ser.closed=true;

clientSocket.close();

}

catch(IOException e)

{

}}}

***Client1:***

import java.io.\*;

import java.net.\*;

public class Client implements Runnable

{

static Socket clientSocket = null;

static PrintStream os = null;

static DataInputStream is = null;

static BufferedReader inputLine = null;

static boolean closed = false;

public static void main(String[] args)

{

int port\_number=1521;

String host="DESKTOP-8KR21VC";

try

{

clientSocket = new Socket(host, port\_number);

inputLine = new BufferedReader(new InputStreamReader(System.in));

os = new PrintStream(clientSocket.getOutputStream());

is = new DataInputStream(clientSocket.getInputStream());

}

catch (Exception e)

{

System.out.println("Exception occurred : "+e.getMessage());

}

if (clientSocket != null && os != null && is != null)

{

try

{

new Thread(new Client()).start();

while (!closed)

{

os.println(inputLine.readLine());

}

os.close();

is.close();

clientSocket.close();

}

catch (IOException e)

{

System.err.println("IOException: " + e);

}

}

} public void run()

{

String responseLine;

try

{

while ((responseLine = is.readLine()) != null)

{

System.out.println("\n"+responseLine);

if (responseLine.equalsIgnoreCase("GLOBAL\_COMMIT")==true || responseLine.equalsIgnoreCase("GLOBAL\_ABORT")==true )

{

break;

}

}

closed=true;

}

catch (IOException e)

{

System.err.println("IOException: " + e);

}}}

***Client2:***

import java.io.\*;

import java.net.\*;

public class Client implements Runnable

{

static Socket clientSocket = null;

static PrintStream os = null;

static DataInputStream is = null;

static BufferedReader inputLine = null;

static boolean closed = false;

public static void main(String[] args)

{

int port\_number=1521;

String host="DESKTOP-8KR21VC";

try

{

clientSocket = new Socket(host, port\_number);

inputLine = new BufferedReader(new InputStreamReader(System.in));

os = new PrintStream(clientSocket.getOutputStream());

is = new DataInputStream(clientSocket.getInputStream());

}

catch (Exception e)

{

System.out.println("Exception occurred : "+e.getMessage());

}

if (clientSocket != null && os != null && is != null)

{

try

{

new Thread(new Client()).start();

while (!closed)

{

os.println(inputLine.readLine());

}

os.close();

is.close();

clientSocket.close();

}

catch (IOException e)

{

System.err.println("IOException: " + e);

}

}

} public void run()

{

String responseLine;

try

{

while ((responseLine = is.readLine()) != null)

{

System.out.println("\n"+responseLine);

if (responseLine.equalsIgnoreCase("GLOBAL\_COMMIT")==true || responseLine.equalsIgnoreCase("GLOBAL\_ABORT")==true )

{

break;

}

}

closed=true;

}

catch (IOException e)

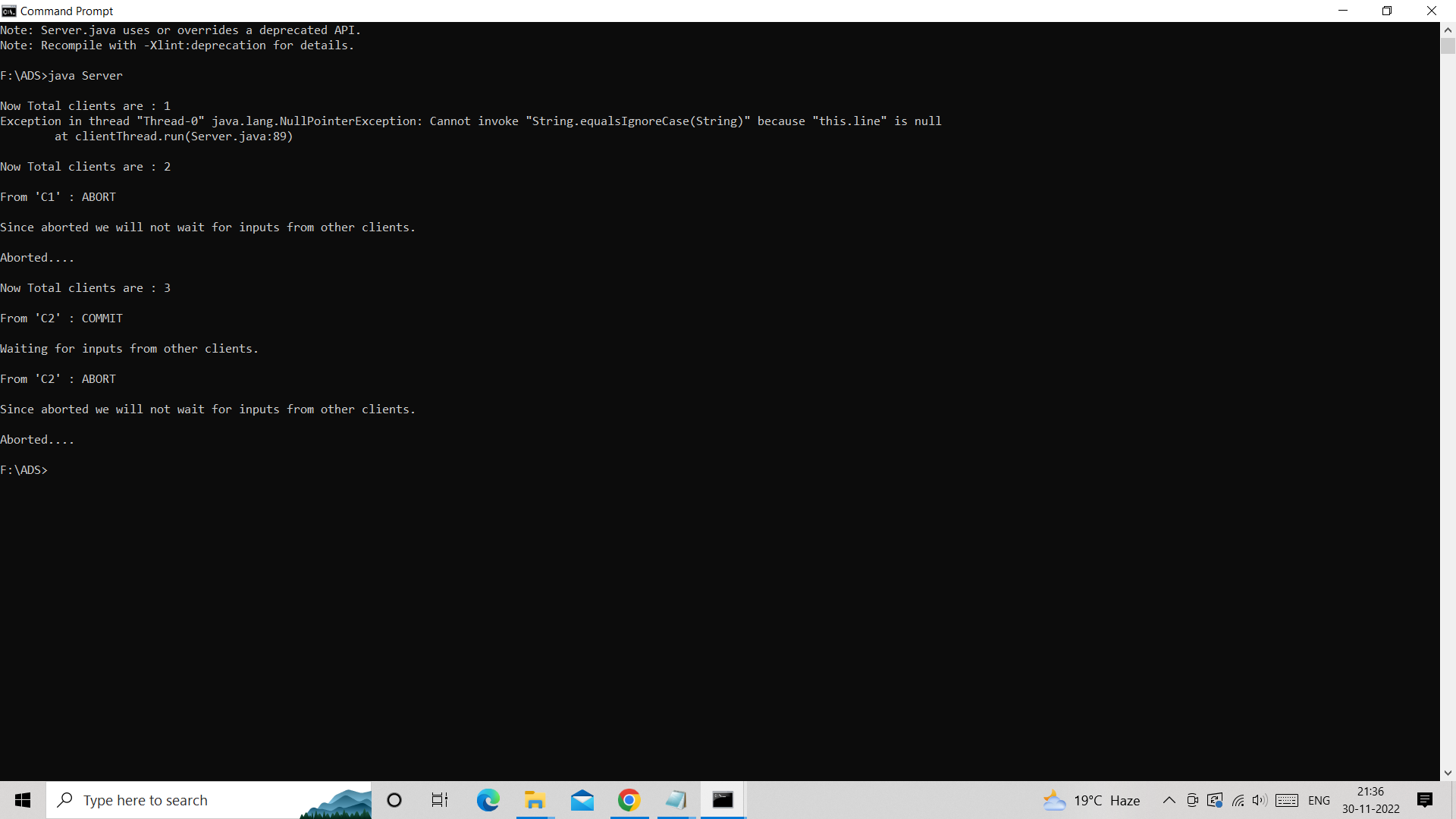
{

System.err.println("IOException: " + e);

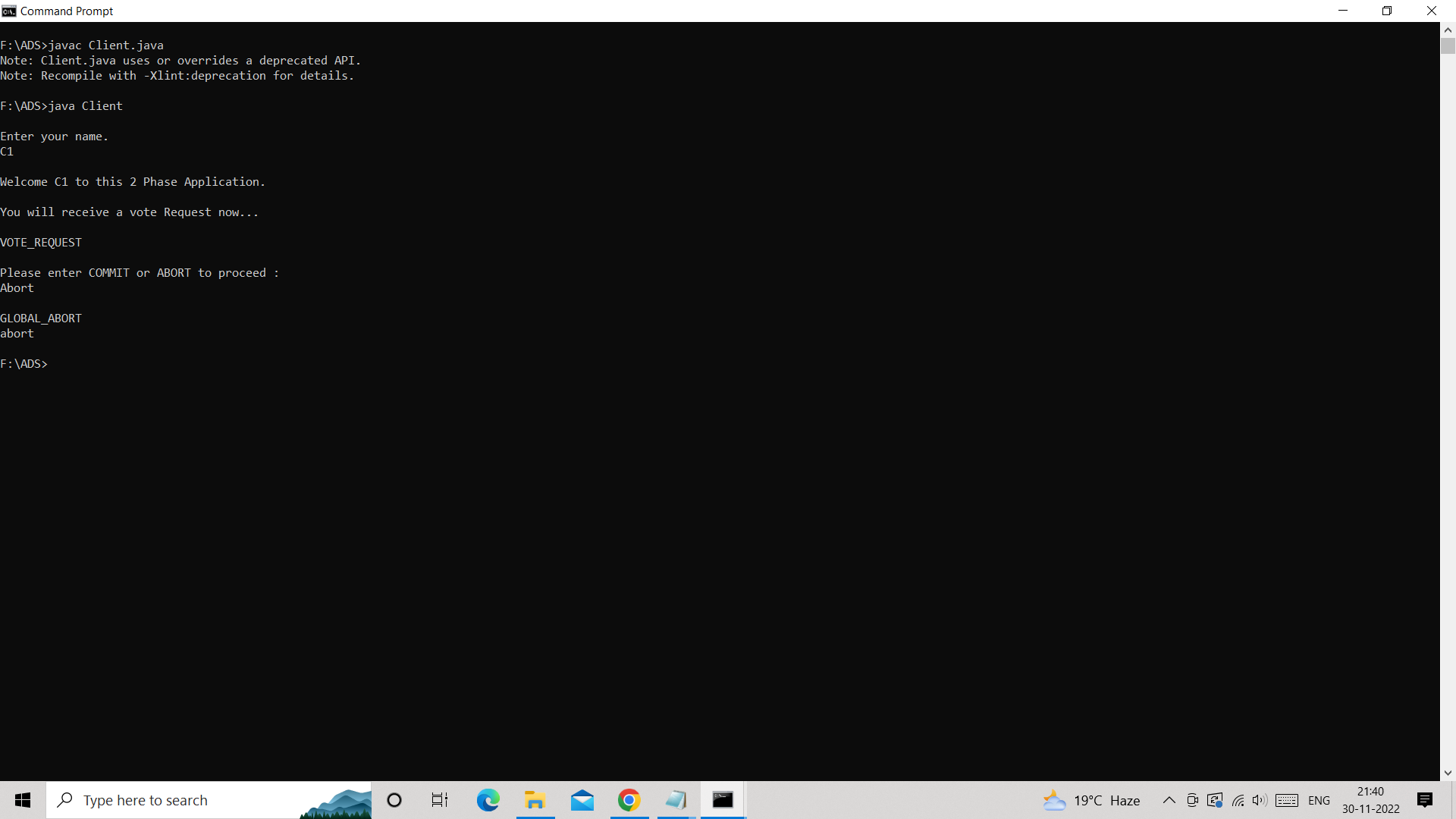
}}}

***Output:***

***Server:***



***Client1:***



***Client2:***

