**PRACTICAL 1**

**Practical 1A: A client server based program using TCP to find if the number entered is prime.**

Step 1: Create two Java files in notepad.

**Code:**

**File 1: tcpClientPrime.java**

import java.net.\*;

import java.io.\*;

class tcpClientPrime

{

public static void main(String args[])

{

try

{

Socket cs = new Socket("LocalHost",8001);

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

System.out.println("Enter a number : ");

int a = Integer.parseInt(infu.readLine());

DataOutputStream out = new

DataOutputStream(cs.getOutputStream());

out.writeInt(a);

DataInputStream in = new DataInputStream(cs.getInputStream()); System.out.println(in.readUTF()); cs.close();

}

catch(Exception e)

{

System.out.println(e.toString());

}

}

}

**File 2: tcpServerPrime.java**

import java.net.\*;

import java.io.\*;

class tcpServerPrime

{

public static void main(String args[])

{

try

{

ServerSocket ss = new ServerSocket(8001);

System.out.println("Server Started...............");

Socket s = ss.accept();

DataInputStream in = new DataInputStream(s.getInputStream()); int x= in.readInt();

DataOutputStream otc = new DataOutputStream(s.getOutputStream()); int y = x/2;

if(x ==1 || x ==2 || x ==3)

{

otc.writeUTF(x + "is Prime");

System.exit(0);

}

for(int i=2; i<=y; i++)

{

if(x%i != 0)

{

otc.writeUTF(x + " is Prime");

}

else

{

otc.writeUTF(x + " is not Prime");

}

}

}

catch(Exception e)

{

System.out.println(e.toString());

}

}

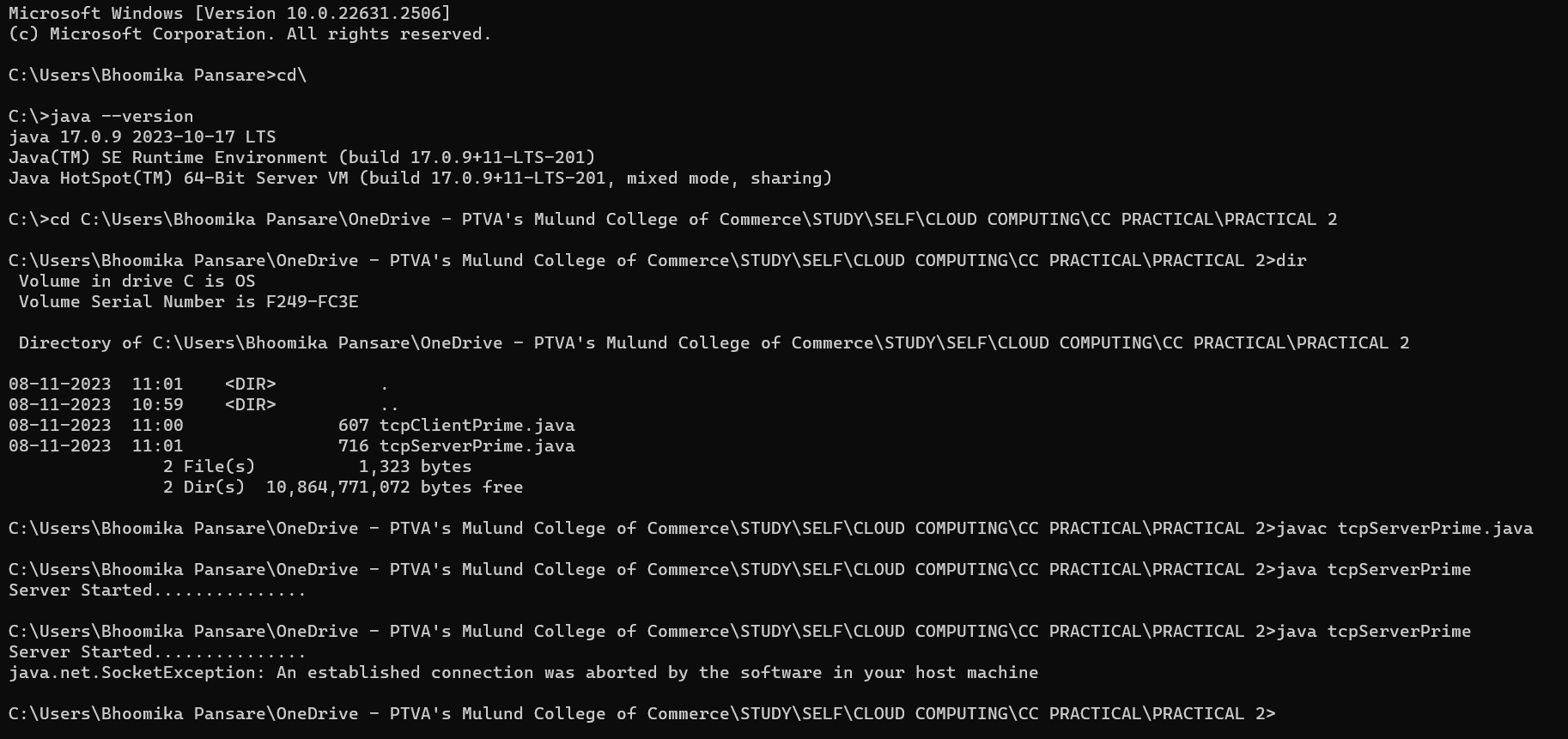
}

Step 2: Open 2 command prompt & enter the following commands to run the files.

In 1st command prompt, enter the following command for Server-side file.

Command 1: javac tcpServerPrime.java

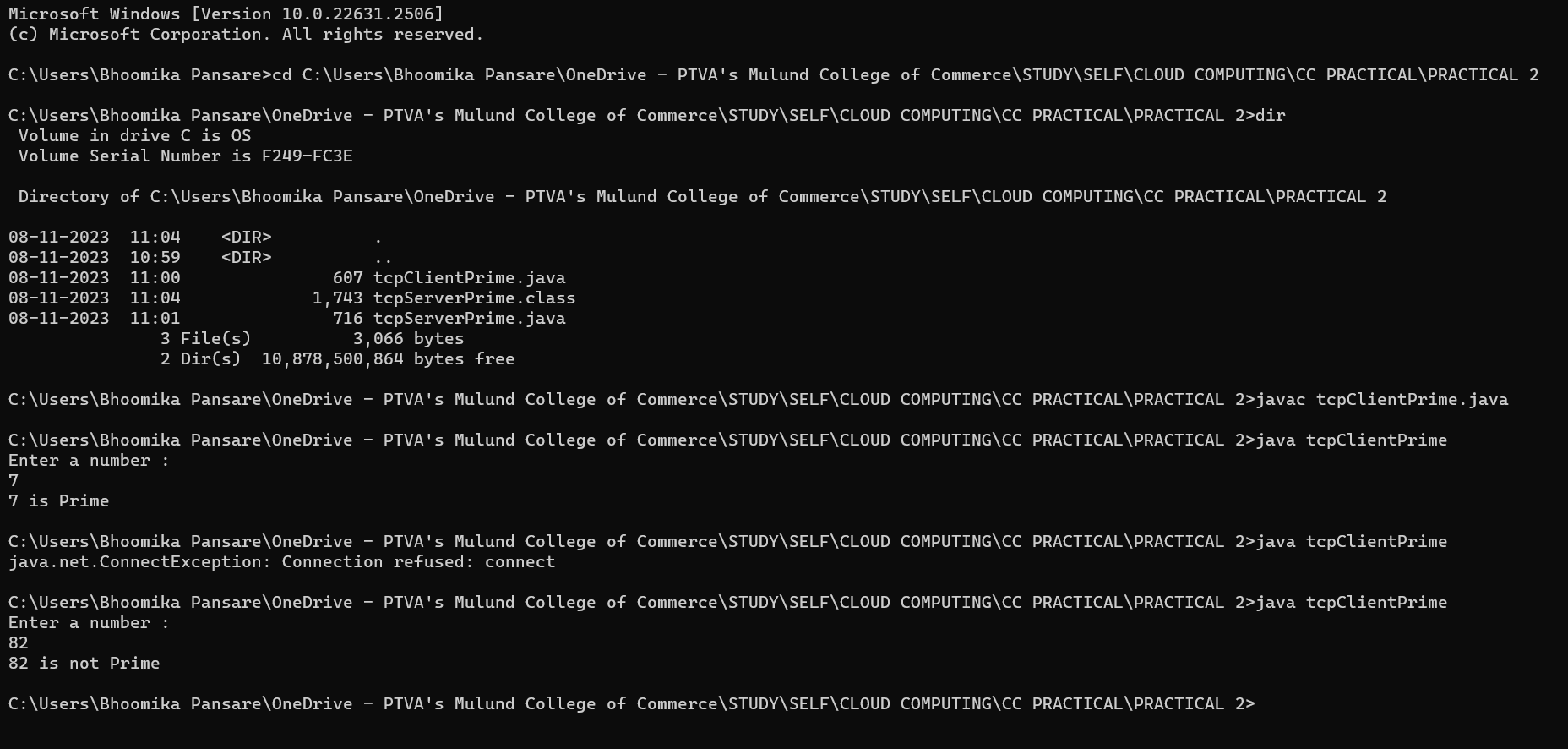
Command 2: java tcpServerPrime



In 2nd command prompt, enter the following command for Client-side file.

Command 1: javac tcpClientPrime.java

Command 2: java tcpClientPrime



**Practical 1B: A client server TCP based chatting application.**

**Code:**

**File 1: ChatServer.java**

import java.net.\*;

import java.io.\*;

class ChatServer

{

public static void main(String args[])

{

try

{

ServerSocket ss = new ServerSocket(8000);

System.out.println("Waiting for client to connect..");

Socket s = ss.accept();

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

DataOutputStream out = new DataOutputStream(s.getOutputStream());

DataInputStream in = new DataInputStream(s.getInputStream());

String receive, send;

while((receive = in.readLine()) != null)

{

if(receive.equals("STOP"))

break;

System.out.println("Client Says : "+receive);

System.out.print("Server Says : ");

send = br.readLine();

out.writeBytes(send+"\n");

}

br.close();

in.close();

out.close();

s.close();

}

catch(Exception e)

{

e.printStackTrace();

}

}

}

**File 2: ChatClient.java**

import java.net.\*;

import java.io.\*;

class ChatClient

{

public static void main(String args[])

{

try

{

Socket s = new Socket("Localhost",8000);

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

DataOutputStream out = new DataOutputStream(s.getOutputStream());

DataInputStream in = new DataInputStream(s.getInputStream());

String msg;

System.out.println("To stop chatting with server type STOP");

System.out.print("Client Says: ");

while((msg = br.readLine()) != null)

{

out.writeBytes(msg+"\n");

if(msg.equals("STOP"))

break;

System.out.println("Server Says : "+in.readLine());

System.out.print("Client Says : ");

}

br.close();

in.close();

out.close();

s.close();

}

catch(Exception e)

{

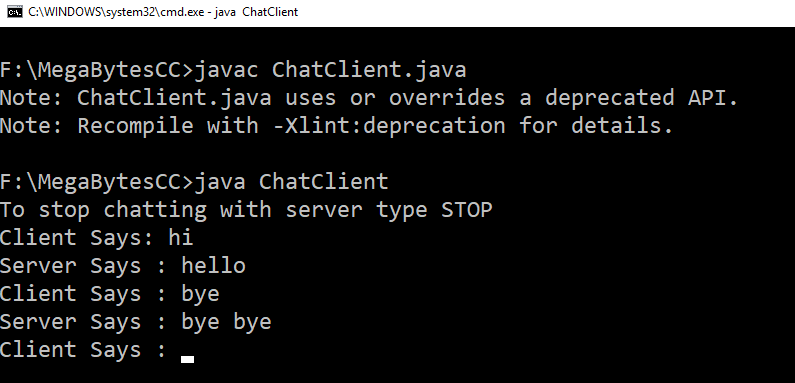
e.printStackTrace();

}

}

}

**Output: Server:**



**Client:**

