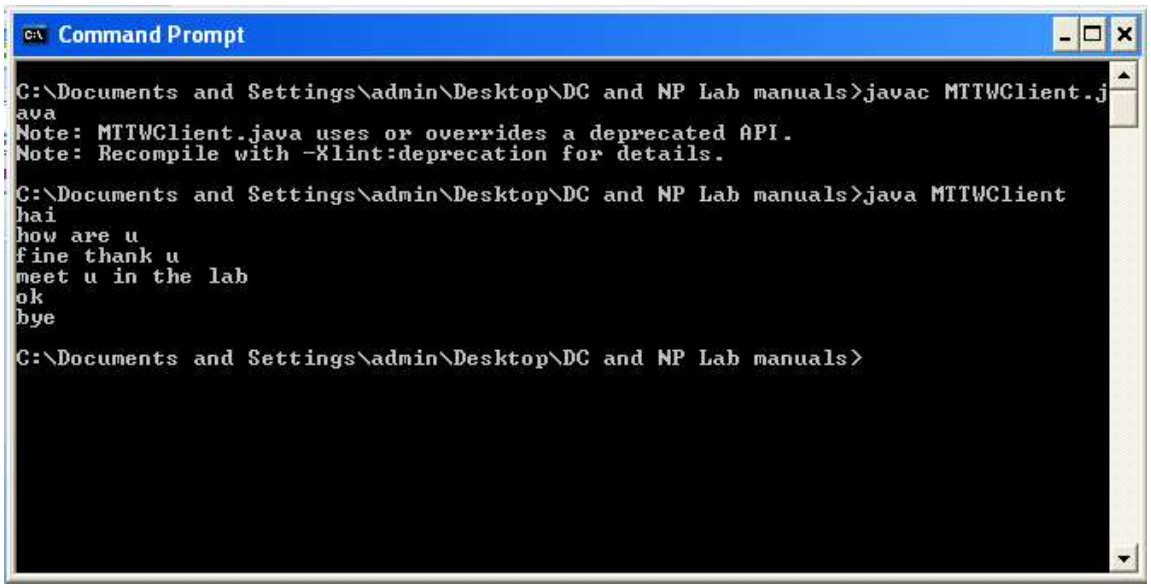


5.5 Implementing Broadcast application

```
import java.io.*;
import java.net.*;
import java.util.*;
class bserver extends Thread
 {
  static ServerSocket ss;
  static Socket cs=null;
  static Vector v=new Vector(100);
  static PrintWriter pw;
  static int id=1;
  public static void main(String args[])
   {
    try
      {
       ss=new ServerSocket(4000);
       InputScan1 is=new InputScan1();
       is.start();
      while(true)
       {
        cs=ss.accept();
        new ClientThread(cs,id).start();
        pw=new PrintWriter(cs.getOutputStream(),true);
         v.addElement(pw);
        System.out.println("connected to client"+id);
        id++;
       }
      }
     catch(Exception e)
```

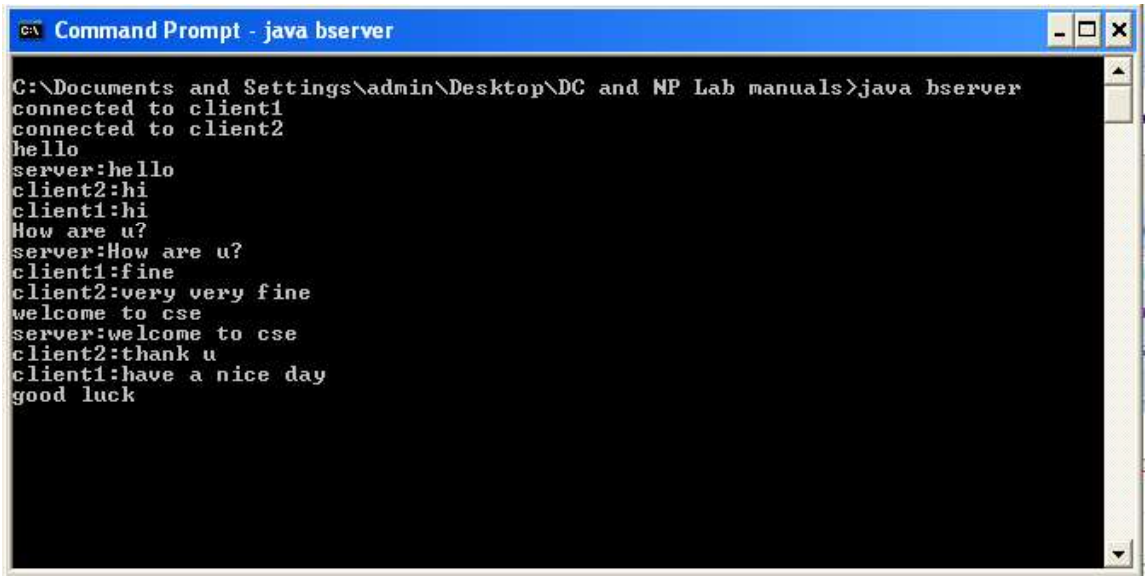
```
            {}
    }
}

class ClientThread extends bserver
{
   Socket cs;
    int id;
    ClientThread(Socket clientsocket,int i)
    {
      cs=clientsocket;
      id=i;
    }

public void run()
 {
      try
    {
     BufferedReader in=new BufferedReader(new InputStreamReader(cs.getInputStream()));
    String msg=null;
     while(true)
      {
      msg=in.readLine();
     System.out.println("client"+id+":"+msg);
      if(msg!=null)
       {
        for(int i=0;i<v.size();i++)
         ((PrintWriter)v.elementAt(i)).println("client"+id+":"+msg);
        }
     }
    }
  catch(Exception e)
    {}
 }
}

class InputScan1 extends bserver
{
   DataInputStream stdin=new DataInputStream(System.in);
   public void run()
   {
     try
       {
       while(true)
            {
             String msg1;
             msg1=stdin.readLine();
             System.out.println("server:"+msg1);
               if(msg1!=null)
                {
```

```
                for(int i=0;i<v.size();i++)
                ((PrintWriter)v.elementAt(i)).println("server"+":"+msg1);
                }
            }
          }
        catch(Exception e)
           { }
    }
}


import java.io.*;
import java.net.*;
public class bclient extends Thread
{
static Socket cs=null;
public static void main(String args[])
{
try
 {
cs=new Socket("localhost",4000);
System.out.println("connected to server");
BufferedReader in=new BufferedReader(new InputStreamReader(cs.getInputStream()));
InputScan is=new InputScan();
is.start();
while(true)
{
String msg;
msg=in.readLine();
System.out.println(msg);
}
}
catch(Exception e)
{
}
}
}

class InputScan extends bclient
{
DataInputStream stdin=new DataInputStream(System.in);
 public void run()
{
try
{
PrintWriter out=new PrintWriter(cs.getOutputStream(),true);
while(true)
{
String msg;
msg=stdin.readLine();
```

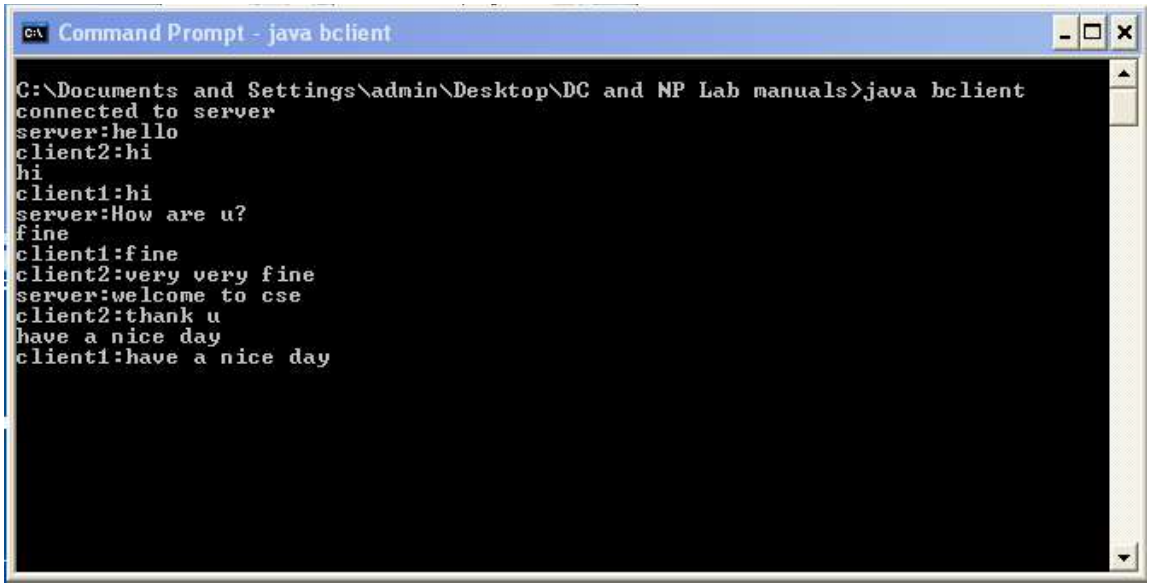
```
out.println(msg);
}
}
catch(Exception e){
}
}
}
```

OUTPUT

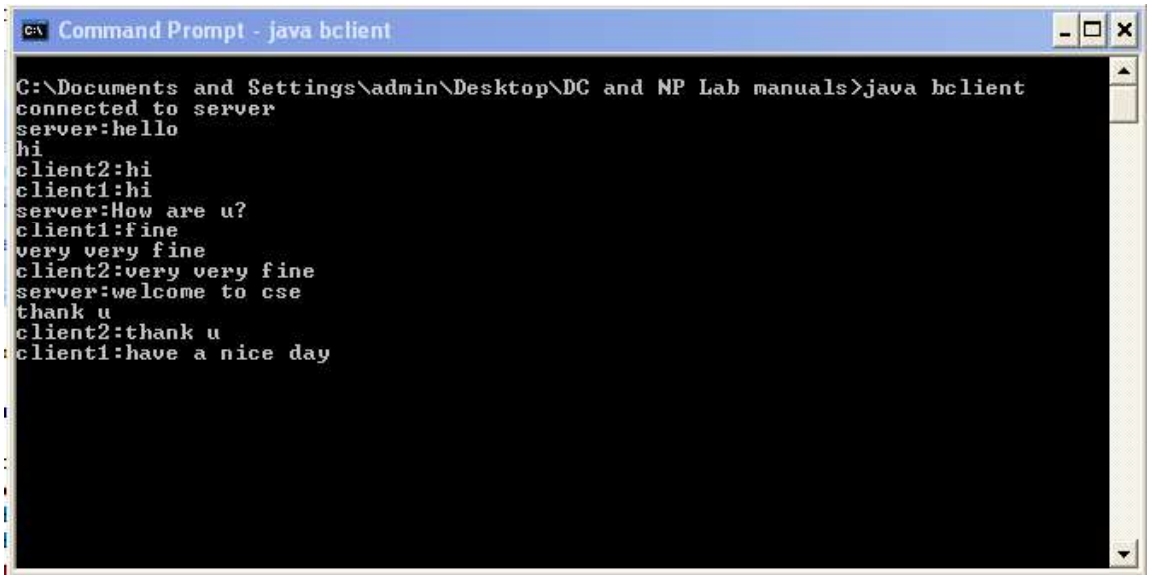
Server



Client 1



Client 2



5.6 Implementing Simple File Transfer Protocol (SFTP)

This program is used to build a server and client to transfer file.

```
import java.io.*;
import java.net.*;
public class SimpleFileServer
{
public static void main(String args[]) throws Exception
{
try
    {
            ServerSocket ss=new ServerSocket(1334);
            Socket s=ss.accept();
             DataInputStream in=new DataInputStream(s.getInputStream());
               DataOutputStream out=new DataOutputStream(s.getOutputStream());
            String str;
             int x=0;
                str=in.readUTF();
                File f=new File(str);
           if(f.exists())
           {  out.write(x);

              FileInputStream fin=new FileInputStream(str);
                int i;
                    do
                            {
                       i=fin.read();
                   out.write(i);
             }
                            while(i!=-1);
```