MULTIUSER CHAT SERVER

AIM

To write a program to implement Multiuser chat server

DESIGN

A chat application consists of a chat server and a chat client. The server accepts connections from the clients and delivers all messages from each client to other clients. This is a tool to communicate with other people over Internet in real time.

The client is implemented using two threads - one thread to interact with the server and the other with the standard input. Two threads are needed because a client must communicate with the server and, simultaneously, it must be ready to read messages from the standard input to be sent to the server.

The server is implemented using threads also. It uses a separate thread for each connection. It spawns a new client thread every time a new connection from a client is accepted

PROGRAM

```
import java.net.*;
import java.io.*;

class pubchatserver
{
  int i,j;
  PrintStream ps[]=new PrintStream[20];
  BufferedReader br[]=new BufferedReader[20];
  Socket s[]=new Socket[20];
  Thread t,t1;
  String str;
  ServerSocket ss;
  pubchatserver()
  {
        try
    {
      ss=new ServerSocket(8000);
      t=new Thread(new ThreadMaker());
         t.start();
    }
  catch(Exception e)
  {

  }
}
class ThreadMaker extends Thread
{
  public void run()
  {
    while(true)
    {
      try
      {
         s[i]=ss.accept();
         ps[i]=new PrintStream(s[i].getOutputStream(),true);
         br[i]=new BufferedReader(new InputStreamReader(s[i].getInputStream()));
               t1=new Thread(new ThreadMaker1(i));
               t1.start();
               i++;
      }
      catch(Exception e)
      {

      }
    }
  }
}
class ThreadMaker1 extends Thread
{
  int n,no;
  ThreadMaker1(int i)
  {
      n=i;
      no=n+1;
  }
  public void run()
  {
    while(true)
    {
      try
```

```
            {
                str=br[n].readLine();
                System.out.println(str);
                        for(j=0;j<=i;j++)
                {
                   if(n!=j)
                     ps[j].println("client"+no+":"+str);
                }
            }
            catch(Exception e)
            {

            }
         }
      }
   }
   public static void main(String args[])
   {
     pubchatserver pcs=new pubchatserver();

   }
}

import java.net.*;
import java.io.*;

class pubchatclient
{

   Socket s;

   BufferedReader br,br1;
   PrintStream ps;
   Thread t1;
   String str,str1;
   pubchatclient()
   {

      try
      {
         s=new Socket("localhost",8000);
            ps=new PrintStream(s.getOutputStream(),true);
            br=new BufferedReader(new InputStreamReader(s.getInputStream()));
           br1=new BufferedReader(new InputStreamReader(System.in));
            t1=new Thread(new ThreadMaker());
            t1.start();
           while(true)
           {
           str=br1.readLine();

          ps.println(str);
}

      }
      catch(Exception e)
      {

      }
    }
    class ThreadMaker extends Thread
    {
      public void run()
      {
        while(true)
        {
          try
          {
             str1=br.readLine();
             System.out.println(str1);
          }
          catch(Exception e)
          {
          }
        }
      }
    }
     public static void main(String args[])
    {
     pubchatclient pcc=new pubchatclient();
        }
```

```
}
```