**PRACTICAL:1**

**AIM :** Create chat application using either TCP or UDP protocol.

**Program:**

Server side program:

import java.net.\*;

import java.io.\*;

class serverchat{

public static void main(String[] args)throws Exception{

ServerSocket serversocket=new ServerSocket(3000);

Socket sc=serversocket.accept();

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

DataInputStream dis=new DataInputStream(new BufferedInputStream(sc.getInputStream()));

DataOutputStream dos=new DataOutputStream(sc.getOutputStream());

String s1="";

String s2="";

while(true){

s1=dis.readUTF();

if(s1.equals("bye")){

System.out.println(s1);

break;

}

System.out.println("client:"+s1);

s2=brs.readLine();

dos.writeUTF(s2);

}

dos.close();

dis.close();

sc.close();

}

Client side program:

import java.net.\*;

import java.io.\*;

class clientchat{

public static void main(String[] args) throws Exception{

Socket socket=new Socket("localhost",3000);

InputStreamReader input=new InputStreamReader(System.in);

BufferedReader br=new BufferedReader(input);

DataInputStream dis=new DataInputStream(new BufferedInputStream(socket.getInputStream()));

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

String s1="";

String s2="";

while(true){

s1=br.readLine();

out.writeUTF(s1);

if((s2=dis.readUTF()) == "bye")

{

System.out.println(s2);

break;

}

System.out.println("server:"+s2);

}

input.close();

out.close();

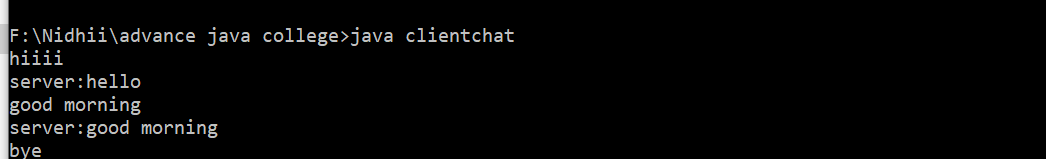
socket.close();

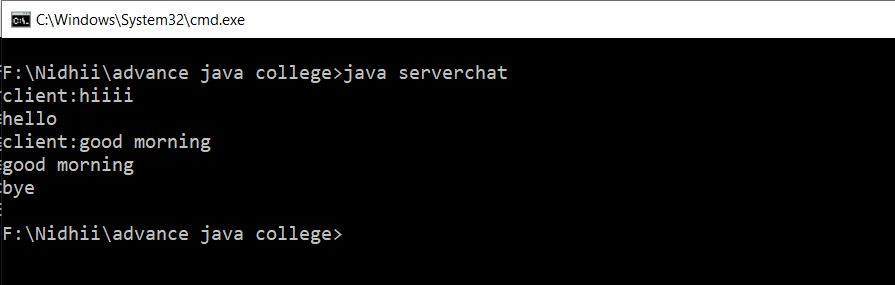
dis.close();

}

}

Output:





**PRACTICAL-2**

**AIM :** Implement any one sorting algorithm using TCP/UDP on Server application and Give Input on Client side and client should sorted output from server and display sorted on input side.

**Program :**

Server side program :

import java.net.\*;

import java.io.\*;

import java.util.Arrays;

class server11{

public static void main(String[] args)throws Exception{

ServerSocket serversocket=new ServerSocket(3000);

Socket sc=serversocket.accept();

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

DataInputStream dis=new DataInputStream(new BufferedInputStream(sc.getInputStream()));

DataOutputStream dos=new DataOutputStream(sc.getOutputStream());

String s1="";

int p=0;

s1=dis.readUTF();

System.out.println(s1);

String[] a=s1.split(" ");

int[] b=new int[a.length];

for(int i=0;i<a.length;i++){

b[i]=Integer.parseInt(a[i]);

}

for(int i=0;i<b.length;i++){

for(int j=0;j<b.length-1;j++){

if(b[j]>b[j+1]){

int tmp=b[j];

b[j]=b[j+1];

b[j+1]=tmp;

}

}

}

String s2 = Arrays.toString(b);

dos.writeUTF(s2);

dos.close();

dis.close();

sc.close();

}

}

Client side program :

import java.net.\*;

import java.io.\*;

class client1{

public static void main(String[] args) throws Exception{

Socket socket=new Socket("localhost",3000);

InputStreamReader input=new InputStreamReader(System.in);

BufferedReader br=new BufferedReader(input);

DataInputStream dis=new DataInputStream(new BufferedInputStream(socket.getInputStream()));

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

String s1="";

String s2="";

while(true){

s1=br.readLine();

out.writeUTF(s1);

if((s2=dis.readUTF()) == "done"){

System.out.println(s2);

break;

}

System.out.println("sorted array:"+s2);

}

input.close();

out.close();

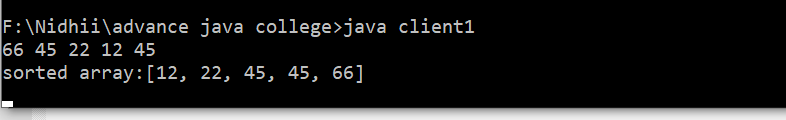
socket.close();

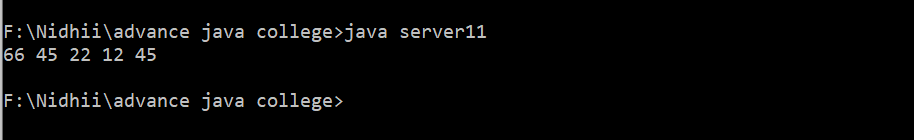
dis.close();

}

}

Output :





**PRACTICAL-3**

**AIM :** Implement Concurrent TCP Server programming in which more than one client can connect and communicate with Server for sending the string and server returns the reverse of string to each of client

**Program :**

Server side program :

import java.net.\*;

import java.io.\*;

class server3{

public static void main(String[] args)throws Exception{

ServerSocket serversocket=new ServerSocket(3000);

Socket sc=null;

while(true){

try{

sc=serversocket.accept();

System.out.println("A new client is connected:"+sc);

DataInputStream dis=new DataInputStream(new BufferedInputStream(sc.getInputStream()));

DataOutputStream dos=new DataOutputStream(sc.getOutputStream());

Thread t=new clienthandler(sc,dis,dos);

t.start();

}

catch(Exception e){

sc.close();

e.printStackTrace();

}

}

}

}

class clienthandler extends Thread

{

final DataInputStream dis;

final DataOutputStream dos;

final Socket s;

public clienthandler(Socket s,DataInputStream dis, DataOutputStream dos)

{

this.s=s;

this.dis=dis;

this.dos=dos;

}

public void run()

{

String received,send;

while(true)

{

try{

received=dis.readUTF();

if(received.equals("exit")){

dos.writeUTF("done");

System.out.println("client:"+s+"wants to"+received);

break;

}

System.out.println("client input:"+received);

StringBuilder sb=new StringBuilder();

sb.append(received);

sb.reverse();

send=sb.toString();

System.out.println("done");

dos.writeUTF(send);

}

catch(Exception e){

e.printStackTrace();

}

}

try{

this.dis.close();

this.dos.close();

}

catch(Exception e){

e.printStackTrace();

}

}

}

Client side program :

import java.net.\*;

import java.io.\*;

class client3{

public static void main(String[] args) throws Exception{

Socket socket=new Socket("localhost",3000);

InputStreamReader input=new InputStreamReader(System.in);

BufferedReader br=new BufferedReader(input);

DataInputStream dis=new DataInputStream(new BufferedInputStream(socket.getInputStream()));

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

String s1="";

String s2="";

while(true){

s1=br.readLine();

out.writeUTF(s1);

if((s2=dis.readUTF()) == "done")

{

System.out.println(s2);

break;

}

System.out.println("reverse:"+s2);

}

input.close();

out.close();

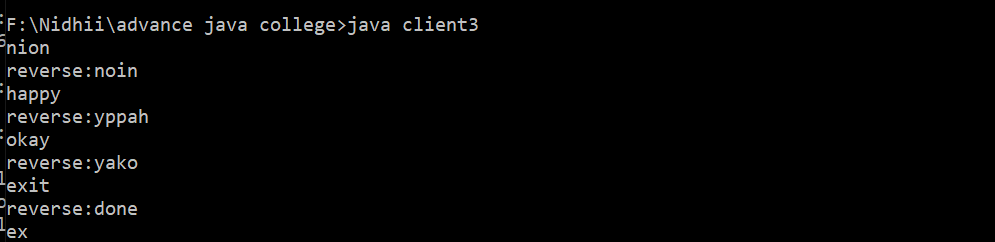
socket.close();

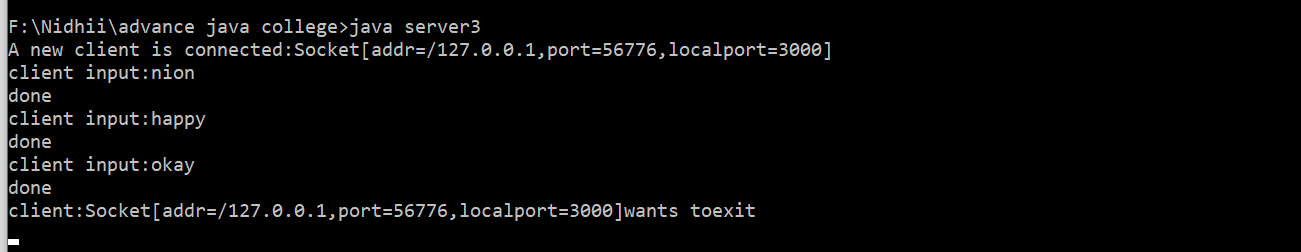
dis.close();

}

}

Output :





**PRACTICAL - 4**

**AIM :** Create Servlet file which contains following functions: 1. Connect 2.Create Database 3. Create Table 4. Insert Records into respective table 5. Update records of particular table of database 6. Delete Records from table. 7. Delete table and also database.

**Program :**

package clg.pracs;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.\*;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class AllDBops

\*/

@WebServlet("/AllDBops")

public class AllDBops extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AllDBops() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

String task = request.getParameter("task");

if (task.equals("DisplayUser")) {

DisplayUser(request,response);

} else if (task.equals("DeleteUser")) {

DeleteUser(request,response);

} else if (task.equals("editForm")) {

EditFormDU(request, response);

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

String task = request.getParameter("task");

if (task.equals("EditFormDBupdate")) {

EditFormDBupdate(request, response);

} else if (task.equals("Reg")) {

RegUser(request,response);

}

}

protected void DisplayUser(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter mPrintWriter = response.getWriter();

mPrintWriter.print("<table align = \"center\" border =\"2px\" width = \"900px\" style=\"background-color: rgb(64,128,128);\r\n" +

" border-collapse : collapse;border: 2px solid rgb(0,0,0);\">\r\n" +

" <tr>\r\n" +

" <th>id</th>\r\n" +

" <th>first\_name</th>\r\n" +

" <th>last\_name</th>\r\n" +

" <th>user\_name</th>\r\n" +

" <th>password</th> \r\n" +

" <th>Action</th> \r\n" +

" </tr>");

try {

Class.forName("com.mysql.jdbc.Driver");

Connection c = DriverManager.getConnection("jdbc:mysql://localhost:3399/practice","root","root");

Statement s = c.createStatement();

java.sql.ResultSet rs = s.executeQuery("Select \* from test");

while(rs.next()){

int id = rs.getInt("id");

String fname = rs.getString("first\_name");

String lname = rs.getString("last\_name");

String uname = rs.getString("user\_name");

String password = rs.getString("password");

mPrintWriter.print("<tr><td>"+id+"</td><td>"+fname+"</td><td>"+lname+"</td><td>"+uname+"</td><td>"+password+"</td><td><a href='AllDBops?task=DeleteUser&value="+id+"'><button>Delete</button></a>&nbsp&nbsp<a href=AllDBops?task=editForm&id="+id+"><button>Edit</button></a></td></tr>");

}

rs.close();

s.close();

c.close();

}catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

protected void DeleteUser(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.getWriter().append("Served at: ").append(request.getContextPath());

int id = Integer.parseInt(request.getParameter("value"));

try {

Class.forName("com.mysql.jdbc.Driver");

Connection c = DriverManager.getConnection("jdbc:mysql://localhost:3399/practice","root","root");

Statement s = c.createStatement();

s.executeUpdate("delete from test where id = '"+id+"'");

s.close();

c.close();

response.sendRedirect("AllDBops?task=DisplayUser");

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

protected void EditFormDU(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html");

int id = Integer.parseInt(request.getParameter("id"));

String fn = "";

String ln = "";

try {

Class.forName("com.mysql.jdbc.Driver");

Connection c = DriverManager.getConnection("jdbc:mysql://localhost:3399/practice","root","root");

Statement s = c.createStatement();

ResultSet rs = s.executeQuery("Select \* from test where id = '"+ id +"' ");

while(rs.next()){

fn = rs.getString("first\_name");

ln = rs.getString("last\_name");

}

rs.close();

s.close();

c.close();

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

PrintWriter out = response.getWriter();

out.print("<form action='AllDBops?task=EditFormDBupdate' method='post'>\r\n" +

"<input type=\"hidden\" name=\"id\" value = "+id+">\r\n" +

"First name : <input type=\"text\" name = \"fname\" value = "+ fn +"> <br>\r\n" +

"Last name : <input type=\"text\" name = \"lname\" value = "+ ln +"> <br>\r\n" +

"<br>\r\n" +

"<input type=\"submit\" name=\"Submit\" value=\"Submit\">\r\n" +

"<br>\r\n" +

"</form>\r\n" +

"");

}

protected void EditFormDBupdate(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

try {

String id = request.getParameter("id");

String fname = request.getParameter("fname");

String lname = request.getParameter("lname");

Class.forName("com.mysql.jdbc.Driver");

Connection c = DriverManager.getConnection("jdbc:mysql://localhost:3399/practice","root","root");

Statement s = c.createStatement();

s.executeUpdate("update test set first\_name = '"+fname+ "' , last\_name = '"+lname+"' where id = '"+ Integer.parseInt(id) +"'");

s.close();

c.close();

response.sendRedirect("AllDBops?task=DisplayUser");

} catch (Exception e) {

// TODO: handle exception

}

}

protected void RegUser(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String fname = request.getParameter("fname");

String lname = request.getParameter("lname");

String uname = request.getParameter("uname");

String password = request.getParameter("password");

try {

Class.forName("com.mysql.jdbc.Driver");

Connection c = DriverManager.getConnection("jdbc:mysql://localhost:3399/practice","root","root");

Statement s = c.createStatement();

int i = s.executeUpdate("insert into test (first\_name,last\_name,user\_name,password) values ('"+fname+"','"+lname+"','"+uname+"','"+password+"')");

s.close();

c.close();

if(i==1){

response.sendRedirect("AllDBops?task=DisplayUser");

}

}catch (ClassNotFoundException | SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

Output:

