# **Name: Abdurrahman Qureshi**

# **Roll No: 210451**

Practical No: 16

**1) Write a program to check credentials of users (Client will send user id and password to server and server will authenticate the client using equals()).**

**CODE:**

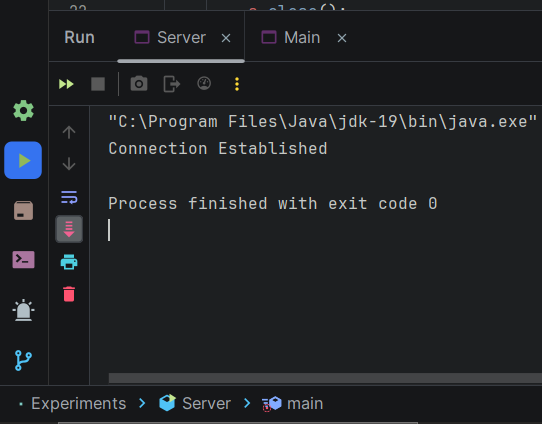
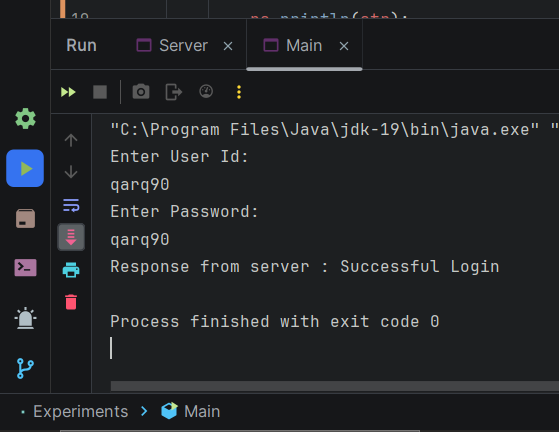
**SERVER**

import java.net.\*;  
import java.io.\*;  
public class Server {  
 public static void main(String args[]) throws Exception {  
 ServerSocket ss = new ServerSocket(777);  
 Socket s = ss.accept();  
 System.*out*.println("Connection Established");  
 OutputStream obj = s.getOutputStream();  
 PrintStream ps = new PrintStream(obj);  
 InputStream obj1 = s.getInputStream();  
 BufferedReader br = new BufferedReader(new InputStreamReader(obj1));  
 String str1 = br.readLine();  
 String str2 = br.readLine();  
 String newstr = "qarq90";  
 if (str1.equals(newstr) && str2.equals(newstr)) {  
 ps.println("Successful Login");  
 } else {  
 ps.println("Invalid Credentials");  
 }  
 ps.close();  
 ss.close();  
 s.close();}}

**CLIENT**

import java.net.\*;  
import java.io.\*;  
public class Main  
{  
 public static void main(String args[]) throws Exception{  
 Socket s=new Socket("localhost",777);  
 BufferedReader kbr=new BufferedReader(new InputStreamReader(System.in));   
 InputStream obj=s.getInputStream();  
 BufferedReader br=new BufferedReader(new InputStreamReader(obj));   
 OutputStream os=s.getOutputStream();  
 PrintStream ps=new PrintStream(os);  
  
 System.out.println("Enter User Id:");  
 String str=kbr.readLine();  
 ps.println(str);  
  
 System.out.println("Enter Password:");  
 String str3=kbr.readLine();  
 ps.println(str3);  
 String newStr=br.readLine();  
 System.out.println("Response from server : "+newStr);  
 br.close();  
 s.close();  
 }}

**OUTPUT:**

**2) Write a program to develop prime number server (Client will send any number to server, Server will send the response the number is prime or not.**

**CODE:**

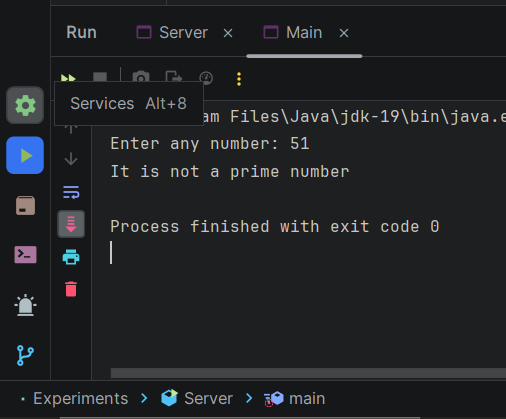
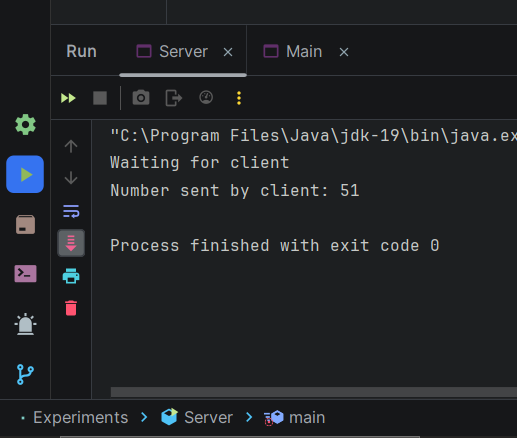
**SERVER**

import java.io.\*;  
import java.net.\*;  
public class Server  
{  
 public static boolean isPrime(int number){  
 boolean isPrimeNum = false;  
 int i = (int) Math.*ceil*(Math.*sqrt*(number));  
 while(i>1)  
 {  
 if((number != i) && (number % i ==0))  
 {  
 isPrimeNum = false;  
 break;  
 }  
 else if(!isPrimeNum)  
 {  
 isPrimeNum = true;  
 }  
 --i;  
 }  
 return isPrimeNum;  
 }  
 public static void main(String [] args) throws Exception  
 {  
 Socket s;  
 int port = 9000;  
 ServerSocket ss = new ServerSocket(port);  
 System.*out*.println("Waiting for client");  
 s = ss.accept();  
 BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));  
 PrintWriter pw = new PrintWriter(new OutputStreamWriter(s.getOutputStream()));  
 int num = Integer.*parseInt*(br.readLine());  
 System.*out*.println("Number sent by client: " + num);  
 pw.println(Server.*isPrime*(num));  
 pw.flush(); }}

**CLIENT**

import java.io.\*;  
import java.net.\*;  
public class Main  
{  
 public static void main(String [] args) throws Exception  
 {  
 int port = 9000;  
 Socket s;  
 BufferedReader br = new BufferedReader(new InputStreamReader(System.*in*));  
 s = new Socket(InetAddress.*getLocalHost*(),port);  
 PrintWriter pw = new PrintWriter(new OutputStreamWriter(s.getOutputStream()));  
 BufferedReader brl = new BufferedReader(new InputStreamReader(s.getInputStream()));  
 System.*out*.print("Enter any number: ");  
 String str = br.readLine();  
 pw.println(str);  
 pw.flush();  
 String msg = brl.readLine();  
 if(msg.equals("true"))  
 {  
 System.*out*.println("It is a prime number");  
 }  
 else  
 {  
 System.*out*.println("It is not a prime number");  
 }  
 }  
}

**OUTPUT:**

**3) Write a program using socket and ServerSocket to create Chat application.**

**CODE:**

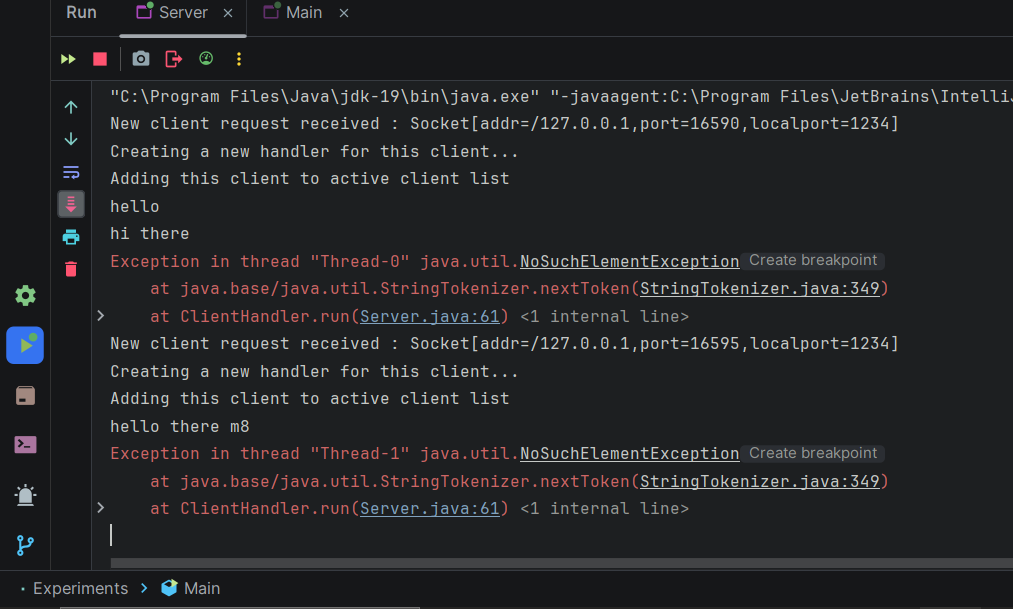
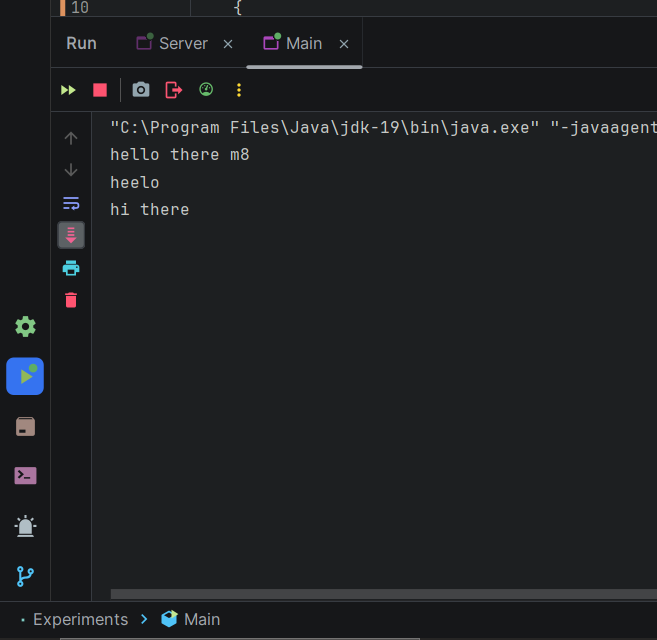
**SERVER**

import java.io.\*;  
import java.util.\*;  
import java.net.\*;  
public class Server  
{  
 static Vector<ClientHandler> *ar* = new Vector<>();  
 static int *i* = 0;  
 public static void main(String[] args) throws IOException{  
 ServerSocket ss = new ServerSocket(1234);  
 Socket s;  
 while (true) {  
 s = ss.accept();  
 System.*out*.println("New client request received : " + s);  
 DataInputStream dis = new DataInputStream(s.getInputStream());  
 DataOutputStream dos = new DataOutputStream(s.getOutputStream());  
 System.*out*.println("Creating a new handler for this client...");  
 ClientHandler mtch = new ClientHandler(s,"client " + *i*, dis, dos);  
 Thread t = new Thread(mtch);  
 System.*out*.println("Adding this client to active client list");  
 *ar*.add(mtch);  
 t.start();  
 *i*++;}}}  
class ClientHandler implements Runnable{  
 Scanner scn = new Scanner(System.*in*);  
 private String name;  
 final DataInputStream dis;  
 final DataOutputStream dos;  
 Socket s;  
 boolean isloggedin;  
 public ClientHandler(Socket s, String name,  
 DataInputStream dis, DataOutputStream dos) {  
 this.dis = dis;  
 this.dos = dos;  
 this.name = name;  
 this.s = s;  
 this.isloggedin=true; }  
 public void run() {  
 String received;  
 while (true) {  
 try{  
 received = dis.readUTF();  
 System.*out*.println(received);  
 if(received.equals("logout")){  
 this.isloggedin=false;  
 this.s.close();  
 break; }  
 StringTokenizer st = new StringTokenizer(received, "#");  
 String MsgToSend = st.nextToken();  
 String recipient = st.nextToken();  
 for (ClientHandler mc : Server.*ar*) {  
 if (mc.name.equals(recipient) && mc.isloggedin==true) {  
 mc.dos.writeUTF(this.name+" : "+MsgToSend);  
 break; }}  
 } catch (IOException e) {  
 e.printStackTrace();}}  
 try{  
 this.dis.close();  
 this.dos.close();  
 }catch(IOException e){  
 e.printStackTrace();}}}

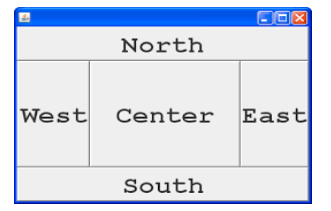
**CLIENT**

import java.io.\*;  
import java.net.\*;  
import java.util.\*;  
  
public class Main  
{  
 final static int *ServerPort* = 1234;  
  
 public static void main(String args[]) throws UnknownHostException, IOException  
 {  
 Scanner scn = new Scanner(System.*in*);  
  
  
 InetAddress ip = InetAddress.*getByName*("localhost");  
  
  
 Socket s = new Socket(ip, *ServerPort*);  
  
  
 DataInputStream dis = new DataInputStream(s.getInputStream());  
 DataOutputStream dos = new DataOutputStream(s.getOutputStream());  
  
  
 Thread sendMessage = new Thread(new Runnable()  
 {  
  
 public void run() {  
 while (true) {  
  
  
 String msg = *scn*.nextLine();  
  
 try {  
  
 *dos*.writeUTF(msg);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 });  
  
  
 Thread readMessage = new Thread(new Runnable()  
 {  
  
 public void run() {  
  
 while (true) {  
 try {  
  
 String msg = *dis*.readUTF();  
 System.*out*.println(msg);  
 } catch (IOException e) {  
  
 e.printStackTrace();  
 }  
 }  
 }  
 });  
  
 sendMessage.start();  
 readMessage.start();  
  
 }  
}

**OUTPUT:**

**4)Generate the following output**

****

**CODE:**

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Main extends Frame {

Main() {

Panel MainBody = new Panel();

MainBody.setLayout(new BorderLayout());

MainBody.add(new Button("NORTH"),BorderLayout.NORTH);

MainBody.add(new Button("SOUTH"),BorderLayout.SOUTH);

MainBody.add(new Button("CENTER"),BorderLayout.CENTER);

MainBody.add(new Button("EAST"),BorderLayout.EAST);

MainBody.add(new Button("WEST"),BorderLayout.WEST);

add(MainBody);

addWindowListener(new java.awt.event.WindowAdapter() {

public void windowClosing(java.awt.event.WindowEvent windowEvent) {

System.exit(0);

}

});

setFont(new Font("Comic Sans MS", Font.BOLD, 20));

setSize(500, 500);

setVisible(true);

setTitle("AJP Mixed Layout");

}

public static void main(String args[]) {

new Main();

}

}

**OUTPUT:**

