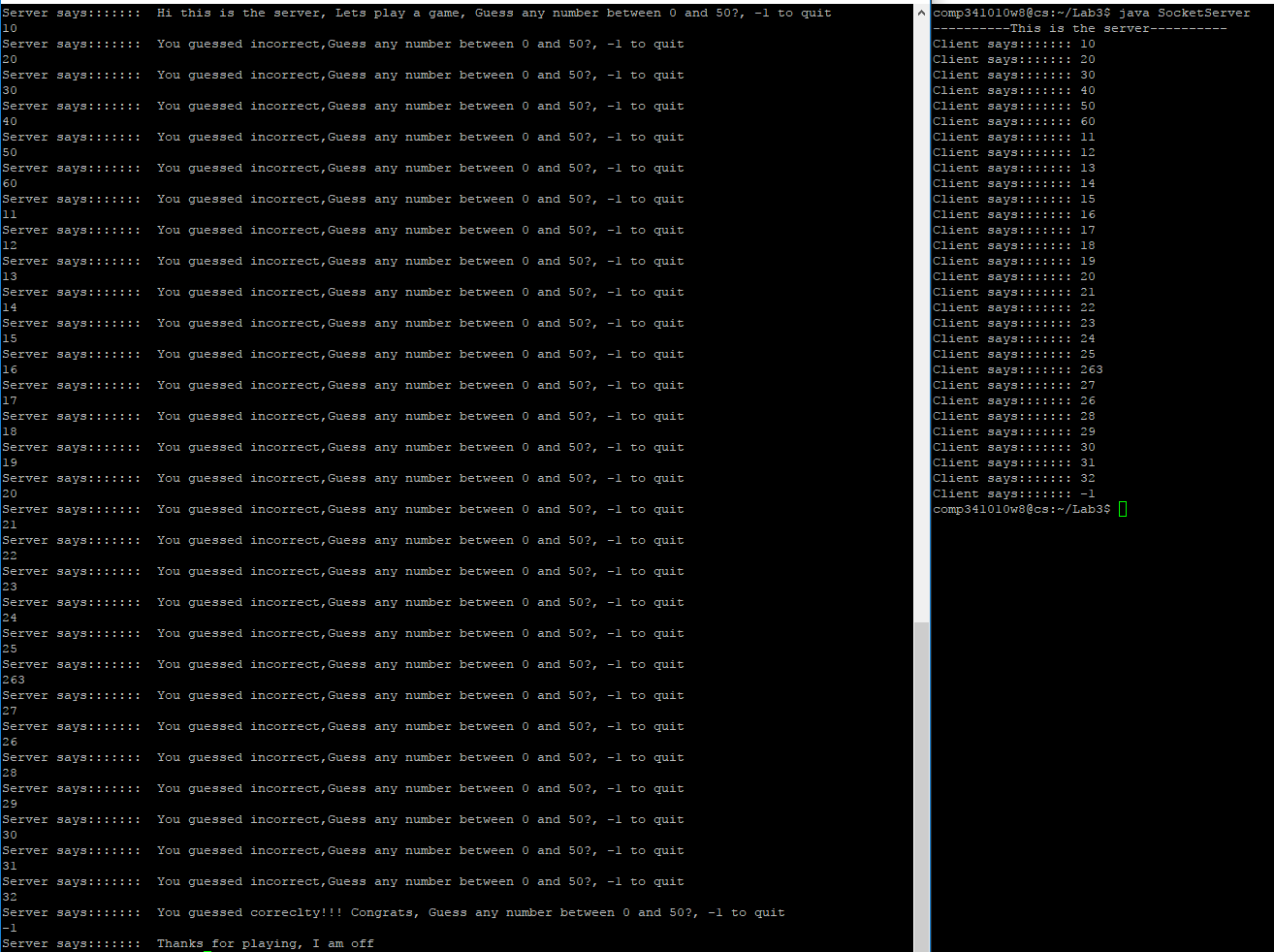
**Output of Client and Server Sessions:**

Jagjit SIngh Bilkhu

T00533766

Comp 3410

IPC via Sockets



**ServerProtocol.java**

**import** java.util.Random;  
*//class to process input provided by client socket and  
// generate random numbers to compare the input***public class** ServerProtocol {  
  
 **private** Random **random**;  
 **private** int generated;

**public** ServerProtocol() {  
 **random** = **new** Random();

generateNumber();  
 }  
 **public void** generateNumber(){  
 **generated = random**.nextInt(50);  
 }  
  
 **public int** processMessage(String line) {  
  
 **try** {  
 **int** x = Integer.*parseInt*(line);  
 **if**(generated==x){  
 **return** 0;  
 }  
 **if**(x<0){  
 **return** 2;  
 }  
 **return** 1;  
 }**catch** (Exception c){  
 **return** -1;  
 }  
 }  
}

**ClientProtocal.java**

**import** java.util.Scanner;  
  
*//Class to read data from keyboard and check returned results from server***public class** ClientProtocol {  
  
 **private static** Scanner *scanner*;  
  
 **public** ClientProtocol(){  
 *scanner* = **new** Scanner(System.***in***);  
 }  
  
 **public** String getInput() {  
  
 String input = *scanner*.nextLine();  
 **return** input;  
 }  
  
}

**SocketServer.java**

**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.io.PrintWriter;  
**import** java.net.ServerSocket;  
**import** java.net.Socket;  
**import** java.net.UnknownHostException;  
**public class** SocketServer {  
 **private static** ServerSocket *serverSocket*;  
 **private static** Socket *socket*;  
 **private static** BufferedReader *bufferedReader*;  
 **private static** PrintWriter *printWriter*;  
 **private static** ServerProtocol *serverProtocol*;  
 **public static void** main(String[] args) {  
 **try** {  
 */\*Establish port to allow client to connect  
 \* Create bufferedReader and PrintWriter to write and read from the client via the socket  
 \* Use clientProtocal to read and process input from client\*/* System.***out***.println(**"----------This is the server----------"**);  
 *serverSocket* = **new** ServerSocket(3010);  
 *socket* = *serverSocket*.accept();  
 *bufferedReader* = **new** BufferedReader(**new** InputStreamReader(*socket*.getInputStream()));  
 *printWriter* = **new** PrintWriter(*socket*.getOutputStream(),**true**);  
 *serverProtocol* = **new** ServerProtocol();  
 *sendMessage*(**"Hi this is the server, Lets play a game, Guess any number between 0 and 50?, -1 to quit"**);  
 **while** (**true**) {  
 String line = *bufferedReader*.readLine();  
 *serverPrintClientMessage*(line);  
 **int** result = *serverProtocol*.processMessage(line);  
 **if**(result==-1){  
 *sendMessage*(**"Sorry I couldnt process that, Guess any number between 0 and 50?, -1 to quit"**);  
 }**else if**(result == 1) {  
 *sendMessage*(**"You guessed incorrect,Guess any number between 0 and 50?, -1 to quit"**);  
 }**else if**(result == 0){

*serverProtocol.generateNumber();*  
 *sendMessage*(**"You guessed correclty!!! Congrats, Guess any number between 0 and 50?, -1 to quit"**);  
 }**else if**(result == 2){  
 *sendMessage*(**"Thanks for playing, I am off"**);  
 *closeAll*();  
 **return**;  
 }  
 }  
 } **catch** (UnknownHostException e) {  
 e.printStackTrace();  
 } **catch** (IOException e) {  
  
 e.printStackTrace();  
 }  
 }  
 */\*Method to print server messages onto the console\*/* **private static void** serverPrintClientMessage( String mesage){  
 System.***out***.println(**"Client says::::::: "**+ mesage);;  
 }  
 */\*Method to send data to client using the printWriter\*/* **private static void** sendMessage(String message){  
 *printWriter*.write(message+**"\n"**);  
 *printWriter*.flush();  
 }  
 */\*Close all sockets and buffers\*/* **private static void** closeAll() **throws** IOException{  
 *printWriter*.close();  
 *bufferedReader*.close();  
 *serverSocket*.close();  
 *socket*.close();  
 }  
}

**SocketClient.java**

**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.io.PrintWriter;  
**import** java.net.Socket;  
**import** java.net.SocketException;  
**import** java.util.Scanner;  
  
*/\*Client class that uses client socket to connect to the server at given port\*/***public class** SocketClient {  
  
  
 **private static** Socket *clientSocket*;  
 **private static** PrintWriter *printWriter*;  
 **private static** BufferedReader *stringBuffer*;  
 **private static** ClientProtocol *clientProtocol*;  
  
 **public static void** main(String[] args) {  
  
 **try** {  
  
 */\*Establish a connection with the server  
 \* Create bufferedReader and PrintWriter to write and read from the server via the socket  
 \* Use clientProtocal to read and process input from server\*/* System.***out***.println(**"----------This is the client----------"**);  
 *clientSocket* = **new** Socket(**"localhost"**,3010);  
 *printWriter* = **new** PrintWriter(*clientSocket*.getOutputStream(),**true**);  
 *stringBuffer* = **new** BufferedReader(**new** InputStreamReader(*clientSocket*.getInputStream()));  
 *clientProtocol* = **new** ClientProtocol();  
  
  
 **try** {  
 **while** (**true**) {  
 *clientPrintServerMessage*(*stringBuffer*.readLine());  
 *sendMessage*(*clientProtocol*.getInput());  
 }  
 }**catch** (SocketException s){  
 System.***out***.println(**"CLOSED"**);  
 *closeAll*();  
 }  
  
 *//closeAll();* } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 */\*Method to print server messages onto the console\*/* **private static void** clientPrintServerMessage( String mesage){  
 System.***out***.println(**"Server says::::::: "**+ mesage);;  
 }  
 */\*Method to send data to client using the printWriter\*/* **private static void** sendMessage(String message){  
 *printWriter*.write(message+**"\n"**);  
 *printWriter*.flush();  
 }  
 */\*Close all sockets and buffers\*/* **private static void** closeAll() **throws** IOException{  
 *printWriter*.close();  
 *stringBuffer*.close();  
 *clientSocket*.close();  
 }  
  
}

**References:**

Sockets programming in Java: A tutorial: Write your own client/server applications seamlessly in Java

<https://www.javaworld.com/article/2077322/core-java/core-java-sockets-programming-in-java-a-tutorial.html>

[Is it possible to run a socket server and socket client on the same machine?](https://stackoverflow.com/questions/10069059/is-it-possible-to-run-a-socket-server-and-socket-client-on-the-same-machine)

<https://stackoverflow.com/questions/10069059/is-it-possible-to-run-a-socket-server-and-socket-client-on-the-same-machine>

Writing the Server Side of a Socket

<https://docs.oracle.com/javase/tutorial/networking/sockets/clientServer.html>