Computer Networks

# Assignment-1

# Question-1:

## Code for Client:

import java.io.\*;

import java.net.\*;

class ClientQ1\_23BCE1145{

private Socket socket = null;

private DataInputStream input = null;

private DataOutputStream output = null;

public ClientQ1\_23BCE1145(String address, int port){

try{

socket = new Socket(address, port);

System.out.println("Client is connected!\n");

input = new DataInputStream(System.in);

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

}

catch(UnknownHostException u){

System.out.println(u);

}

catch(IOException i){

System.out.println(i);

}

}

@SuppressWarnings("deprecation")

public void Work(){

String line = "";

while(!line.equalsIgnoreCase("end")){

try{

line = input.readLine();

output.writeUTF(line);

}

catch(IOException i){

System.out.println(i);

}

}

try{

input.close();

output.close();

socket.close();

}

catch(IOException i){

System.out.println(i);

}

}

public static void main(String[] args){

ClientQ1\_23BCE1145 client = new ClientQ1\_23BCE1145("127.0.0.1", 5000);

client.Work();

}

}

## Code for Server:

import java.io.\*;

import java.net.\*;

public class ServerQ1\_23BCE1145 {

private ServerSocket server = null;

private Socket socket = null;

private DataInputStream input = null;

public ServerQ1\_23BCE1145(int port){

try{

System.out.println("Starting server.\n");

server = new ServerSocket(port);

System.out.println("Waiting for a connection.\n");

socket = server.accept();

System.out.println("Connection with client established.\n");

input = new DataInputStream(socket.getInputStream());

}

catch(UnknownHostException u){

System.out.println(u);

}

catch(IOException i){

System.out.println(i);

}

}

public void Work(){

String line = "";

while(!line.equalsIgnoreCase("end")){

try{

line = input.readUTF();

System.out.println(line);

}

catch(IOException i){

System.out.println(i);

}

}

System.out.println("\nClosing Connection.");

try{

input.close();

socket.close();

server.close();

}

catch(IOException i){

System.out.println(i);

}

}

public static void main(String[] args){

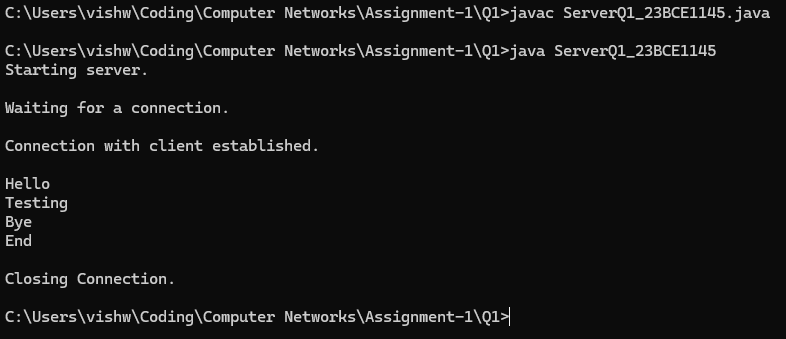
ServerQ1\_23BCE1145 server = new ServerQ1\_23BCE1145(5000);

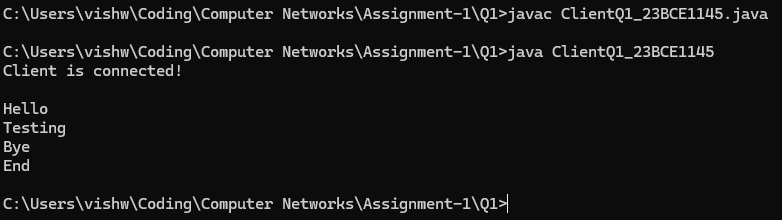
server.Work();

}

}

## Outputs:





# Question-2:

## Code for Client:

import java.io.\*;

import java.net.\*;

public class ClientQ2\_23BCE1145 {

private Socket socket = null;

private DataInputStream input = null;

private DataOutputStream output = null;

public ClientQ2\_23BCE1145(String address, int port){

try{

socket = new Socket(address, port);

System.out.println("Client is connected!\n\nType 'End' to stop the programs.\n");

input = new DataInputStream(System.in);

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

}

catch(UnknownHostException u){

System.out.println(u);

}

catch(IOException i){

System.out.println(i);

}

}

@SuppressWarnings("deprecation")

public void Work(){

String line = "";

while(!line.equalsIgnoreCase("end")){

try{

line = input.readLine();

output.writeUTF(line);

}

catch(IOException i){

System.out.println(i);

}

}

try{

System.out.println("\nClosing the client.");

input.close();

output.close();

socket.close();

}

catch(IOException i){

System.out.println(i);

}

}

public static void main(String[] args){

ClientQ2\_23BCE1145 client = new ClientQ2\_23BCE1145("127.0.0.1", 5000);

client.Work();

}

}

## Code for Server:

import java.io.\*;

import java.net.\*;

public class ServerQ2\_23BCE1145 {

private ServerSocket server = null;

private Socket socket = null;

private DataInputStream input = null;

public ServerQ2\_23BCE1145(int port){

try{

System.out.println("Starting the server.\n");

server = new ServerSocket(port);

System.out.println("Waiting for a client request.\n");

socket = server.accept();

System.out.println("Connected to a client.\n");

input = new DataInputStream(socket.getInputStream());

}

catch(UnknownHostException u){

System.out.println(u);

}

catch(IOException i){

System.out.println(i);

}

}

public void Check(Integer n){

int i;

for(i = 2; i < n; i++){

if(n % i == 0){

System.out.println(n + " is not a prime number.");

return;

}

}

if(n == 1){

System.out.println("1 is neither prime nor composite.");

}

else if(n <= 0){

System.out.println(n + " is not a natural number.");

}

else{

System.out.println(n + " is a prime number.");

}

}

public void Work(){

String line = "";

while(!line.equalsIgnoreCase("end")){

try{

line = input.readUTF();

Check(Integer.parseInt(line));

}

catch(IOException i){

System.out.println(i);

}

catch(NumberFormatException n){

if(line.equalsIgnoreCase("End")){

break;

}

System.out.println("Input a number, and nothing else.");

}

}

try{

System.out.println("\nClosing the server.");

input.close();

socket.close();

server.close();

}

catch(IOException i){

System.out.println(i);

}

}

public static void main(String[] args){

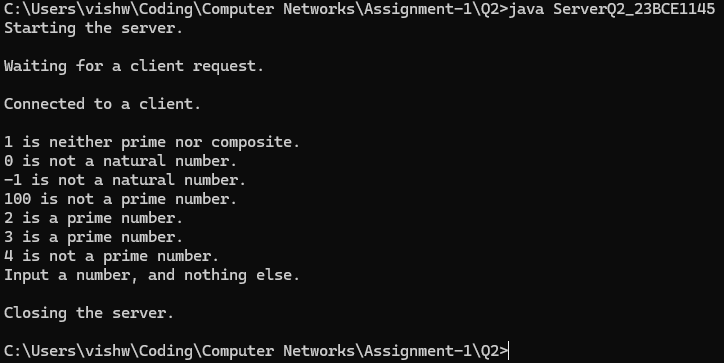
ServerQ2\_23BCE1145 server = new ServerQ2\_23BCE1145(5000);

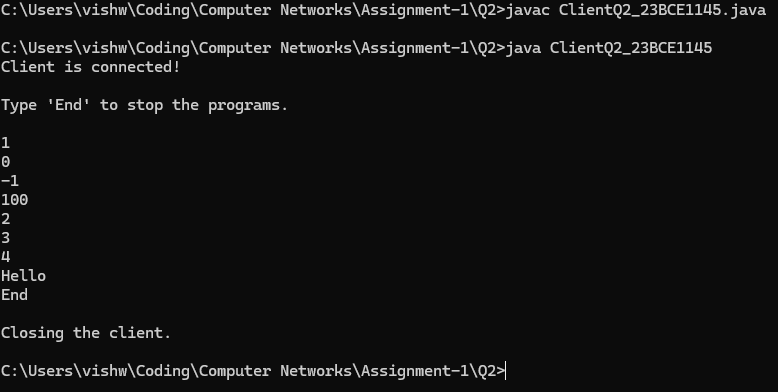
server.Work();

}

}

## Outputs:





# Question-3:

## Code for Client:

import java.io.\*;

import java.net.\*;

public class ClientQ3\_23BCE1145 {

private Socket socket = null;

private DataInputStream input = null;

private DataOutputStream output = null;

private DataInputStream input2 = null;

public ClientQ3\_23BCE1145(String address, int port){

try{

socket = new Socket(address, port);

System.out.println("Client is connected!\n\nType 'End' to stop the programs.\n");

input = new DataInputStream(System.in);

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

input2 = new DataInputStream(socket.getInputStream());

}

catch(UnknownHostException u){

System.out.println(u);

}

catch(IOException i){

System.out.println(i);

}

}

@SuppressWarnings("deprecation")

public void Work(){

String line = "", l = "";

while(!line.equalsIgnoreCase("end")){

try{

line = input.readLine();

output.writeUTF(line);

l = input2.readUTF();

System.out.println(l);

}

catch(IOException i){

System.out.println(i);

}

}

try{

System.out.println("\nClosing the client.");

input.close();

output.close();

socket.close();

}

catch(IOException i){

System.out.println(i);

}

}

public static void main(String[] args){

ClientQ3\_23BCE1145 client = new ClientQ3\_23BCE1145("127.0.0.1", 5000);

client.Work();

}

}

## Code for Server:

import java.io.\*;

import java.net.\*;

public class ServerQ3\_23BCE1145 {

private ServerSocket server = null;

private Socket socket = null;

private DataInputStream input = null;

private DataOutputStream output = null;

public ServerQ3\_23BCE1145(int port){

try{

System.out.println("Starting the server.\n");

server = new ServerSocket(port);

System.out.println("Waiting for a client request.\n");

socket = server.accept();

System.out.println("Connected to a client.\n");

input = new DataInputStream(socket.getInputStream());

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

}

catch(UnknownHostException u){

System.out.println(u);

}

catch(IOException i){

System.out.println(i);

}

}

public void Check(Integer n){

int i;

try{

for(i = 2; i < n; i++){

if(n % i == 0){

output.writeUTF(n + " is not a prime number.");

return;

}

}

if(n == 1){

output.writeUTF("1 is neither prime nor composite.");

}

else if(n <= 0){

output.writeUTF(n + " is not a natural number.");

}

else{

output.writeUTF(n + " is a prime number.");

}

}

catch(IOException i1){

System.out.println(i1);

}

}

public void Work(){

String line = "";

while(!line.equalsIgnoreCase("end")){

try{

line = input.readUTF();

Check(Integer.parseInt(line));

System.out.println(line);

}

catch(IOException i){

System.out.println(i);

}

catch(NumberFormatException n){

if(line.equalsIgnoreCase("End")){

break;

}

try{

System.out.println(line);

output.writeUTF("Input a number, and nothing else.");

}

catch(IOException i1){

System.out.println(i1);

}

}

}

try{

System.out.println("\nClosing the server.");

input.close();

socket.close();

server.close();

}

catch(IOException i){

System.out.println(i);

}

}

public static void main(String[] args){

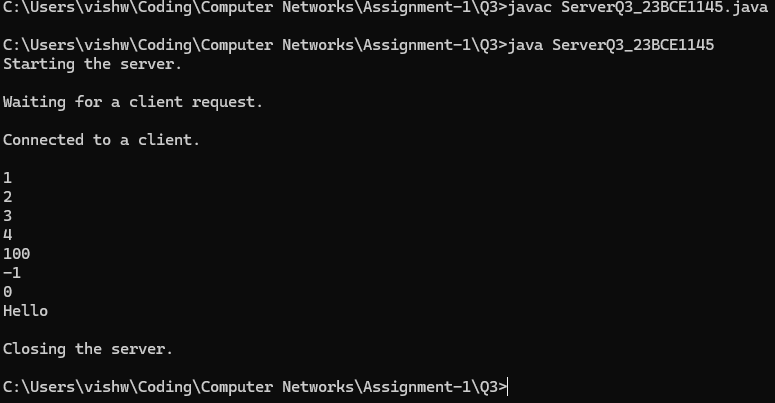
ServerQ3\_23BCE1145 server = new ServerQ3\_23BCE1145(5000);

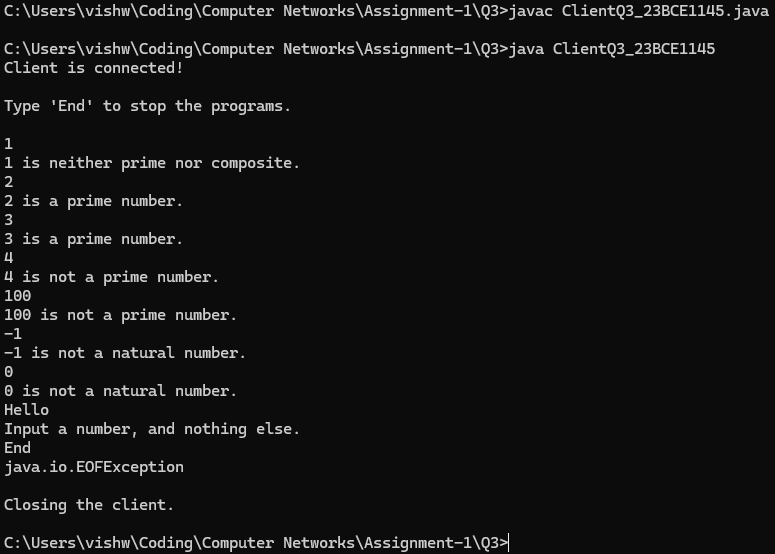
server.Work();

}

}

## Outputs:





### Done by:

#### S Vishwajith

#### 23BCE1145