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

public class RemoteVariableClientUDP{
    static InetAddress aHost;
    static int serverPort;
    static DatagramSocket aSocket;
    static String replyString = "";


    public static void main(String args[]){
        // args give message contents and server hostname
        System.out.println("The client is running.");
        try {
            // set to localhost to host on local machine and set port
            aHost = InetAddress.getByName("localhost");
            serverPort = 6789;

            // input server port to use
            System.out.print("Input a server side port number: ");
            Scanner readline = new Scanner(System.in);
            serverPort = readline.nextInt();


            String ID = "";
            String operation = "";
            String value = "";

            String packet = "";

            while (true) {
                System.out.println("\n1. Add a value to your sum.\n" +
                        "2. Subtract a value from
```

```
your sum.\n" +
                        "3. Get your sum.\n" +
                        "4. Exit client");

                int in = readline.nextInt();

                if(in == 1){
                    operation = "1 ";
                    System.out.println("Enter value
 to add: ");
                    value = String.valueOf(readline
.nextInt());
                    System.out.println("Enter your
ID: ");
                    ID = String.valueOf(readline.
nextInt()) + " ";

                    packet += operation + ID +
value;

                }
                else if(in == 2){
                    operation = "2 ";
                    System.out.println("Enter value
 to subtract: ");
                    value = String.valueOf(readline
.nextInt());
                    System.out.println("Enter your
ID: ");
                    ID = String.valueOf(readline.
nextInt()) + " ";

                    packet += operation + ID +
value;

                }
                else if(in == 3){
                    operation = "3 ";
                    System.out.println("Enter your
ID: ");
                    ID = String.valueOf(readline.
```

```java
nextInt()) + " ";
                    value = "";

                    packet += operation + ID + value;
                }
                else if(in == 4){
                    System.out.println("Client side quitting. The remote variable server is still running."); // send reply if not halt
                    break;
                }

                System.out.println("The result is : " + communicate(packet));

                packet = "";

            }

            // catch potential exceptions
        }catch (SocketException e) {System.out.println("Socket: " + e.getMessage());
        }catch (IOException e){System.out.println("IO: " + e.getMessage());
        }finally {if(aSocket != null) aSocket.close();}
    }
    public static String communicate(String in) throws IOException {

        aSocket = new DatagramSocket();

        // operation code
        byte [] m = in.getBytes(); // convert console in into byte packets
        DatagramPacket operationRequest = new DatagramPacket(m,  m.length, aHost, serverPort);
        aSocket.send(operationRequest); // send packet

        byte[] buffer = new byte[1000]; // create a buffer to receiver server packet
        DatagramPacket reply = new DatagramPacket(buffer, buffer.length); // format for receiving packet
        aSocket.receive(reply); // receive from socket
        replyString = new String(reply.getData()).substring(0,reply.getLength()); // get proper length of string

        return replyString;
    }
}
```

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

public class RemoteVariableServerUDP{
    private static int sum = 0;
    private static TreeMap<Integer, Integer> tree_map = new TreeMap<Integer, Integer>();

    static String instructions = "1. Add a value to your sum.\n" +
            "2. Subtract a value from your sum.\n" +
            "3. Get your sum.\n" +
            "4. Exit client";

    public static void main(String args[]){
        System.out.println("The server is running."); // lab instructions
        DatagramSocket aSocket = null;
        byte[] buffer = new byte[1000];  // set up packet buffer for client message
        try{
            // set up ports
            System.out.print("Input a server port number to listen on: ");
            Scanner readline = new Scanner(System.in);
            int serverPort = readline.nextInt(); // convert to int

            // set up sockets to receive client packets
            aSocket = new DatagramSocket(serverPort);
            aSocket.setReuseAddress(true);

            DatagramPacket request = new DatagramPacket(buffer, buffer.length); // syntax for buffer

            int id;
            int value;

            while(true){ // loop to continue until 'halt!' is sent

                // operation request and reply
                aSocket.receive(request); // receive request, and format a reply
                String requestString = new String(request.getData()).substring(0,request.getLength()).strip(); // proper length

                String[] arrRequestString = requestString.split(" ");

                id = Integer.valueOf(arrRequestString[1]);

                if(!tree_map.containsKey(id)){
                    if(arrRequestString[0].equals("1")){
                        value = Integer.valueOf(arrRequestString[2]);
                        tree_map.put(id, value);
                    }
                    else if(arrRequestString[0].equals("2")){
                        value = -1*Integer.valueOf(arrRequestString[2]);
                        tree_map.put(id, value);
                    }
                }
                else{
                    if(arrRequestString[0].equals("1")){
                        value = Integer.valueOf(arrRequestString[2]);
                        tree_map.replace(id, tree_map.get(id)+value);
```

```java
          }
          else if(arrRequestString[0].equals("2")){
            value = Integer.valueOf(arrRequestString[2]);
            tree_map.replace(id, tree_map.get(id)-value);
          }
        }

        System.out.println("\nVisitorID: " + arrRequestString[1]);
        System.out.println("Operation #: " + arrRequestString[0]);
        System.out.println("Return Variable " + String.valueOf(tree_map.get(id)));


        byte [] m = (String.valueOf(tree_map.get(id))).getBytes(); // convert console in into byte packets

        DatagramPacket reply  = new DatagramPacket(m,  m.length, request.getAddress(), request.getPort());
        aSocket.send(reply); // send back the reply


      }
      // catch potential exceptions
    }catch (SocketException e){System.out.println("Socket: " + e.getMessage());
    }catch (IOException e) {System.out.println("IO: " + e.getMessage());
    }finally {if(aSocket != null) aSocket.close();}
  }
}
```