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

public class AddingClientUDP{
    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 nextLine;
            BufferedReader typed = new BufferedReader(new InputStreamReader(System.in));
            while ((nextLine = typed.readLine()) != null) {
                // halt logic
                if(nextLine.equalsIgnoreCase("halt!")){
                    System.out.println("Client side quitting.");
                    break;
```

```
32                  }
33                  else{
34                      int result = add(Integer.
   parseInt(nextLine));
35                      System.out.println("The server
   returned: " + replyString + "."); // send reply if
   not halt
36                  }
37              }
38
39              // catch potential exceptions
40          }catch (SocketException e) {System.out.
   println("Socket: " + e.getMessage());
41          }catch (IOException e){System.out.println("
   IO: " + e.getMessage());
42          }finally {if(aSocket != null) aSocket.close
   ();}
43      }
44      public static int add(int i) throws IOException
    {
45          // set up network socket to allow
   communication between server and client
46          aSocket = new DatagramSocket();
47
48          byte [] m = String.valueOf(i).getBytes();
   // convert console in into byte packets
49
50          DatagramPacket request = new DatagramPacket
   (m,  m.length, aHost, serverPort);
51          aSocket.send(request); // send packet
52
53          byte[] buffer = new byte[1000]; // create a
    buffer to receiver server packet
54          DatagramPacket reply = new DatagramPacket(
   buffer, buffer.length); // format for receiving
   packet
55          aSocket.receive(reply); // receive from
   socket
56          replyString = new String(reply.getData()).
   substring(0,reply.getLength()); // get proper
   length of string
```

```
57
58          return Integer.parseInt(replyString);
59      }
60  }
```

```
 1 import java.net.*;
 2 import java.io.*;
 3 import java.util.Scanner;
 4
 5 public class AddingServerUDP{
 6     private static int sum = 0;
 7
 8     public static void main(String args[]){
 9         System.out.println("The server is running."
   ); // lab instructions
10         DatagramSocket aSocket = null;
11         byte[] buffer = new byte[1000];  // set up
   packet buffer for client message
12         try{
13             // set up ports
14             System.out.print("Input a server port
   number to listen on: ");
15             Scanner readline = new Scanner(System.
   in);
16             int serverPort = readline.nextInt();
   // convert to int
17
18             // set up sockets to receive client
   packets
19             aSocket = new DatagramSocket(serverPort
   );
20             aSocket.setReuseAddress(true);
21
22             DatagramPacket request = new
   DatagramPacket(buffer, buffer.length); // syntax
   for buffer
23             while(true){ // loop to continue until
    'halt!' is sent
24                 aSocket.receive(request); //
   receive request, and format a reply
25
26                 String requestString = new String(
   request.getData()).substring(0,request.getLength
   ()); // proper length
27
28                 System.out.println("Adding " +
```

```
28 requestString + " to " + String.valueOf(sum));
29
30                 sum += Integer.valueOf(
   requestString.toString());
31
32                 String replyString = String.valueOf
   (sum);
33
34                 byte [] m = replyString.getBytes
   (); // convert console in into byte packets
35                 DatagramPacket reply  = new
   DatagramPacket(m,  m.length, request.getAddress(),
   request.getPort());
36
37                 System.out.println("Returning sum
   of "+replyString + " to client.\n");
38                 aSocket.send(reply); // send back
   the reply
39
40                 // halt logic
41                 if(replyString.equalsIgnoreCase("
   halt!")){
42                     System.out.print("Server side
   quitting");
43                     break;
44                 }
45             }
46             // catch potential exceptions
47         }catch (SocketException e){System.out.
   println("Socket: " + e.getMessage());
48         }catch (IOException e) {System.out.println(
   "IO: " + e.getMessage());
49         }finally {if(aSocket != null) aSocket.close
   ();}
50     }
51 }
```