**In2cm Report**

**How to run the program**

The program can be run by first compiling the server and client. Next you run the server and then you can use the client. The client format will be In2cmClient [inches] and In2cmClient2 [inches] [host name]. There is no difference between In2cmServer and In2cmServer2 so you can use either for testing the client. Make sure to close the server after you are done using it (All sockets except server socket are terminated in the program). Running the program in eclipse is shown in the screenshots section below.

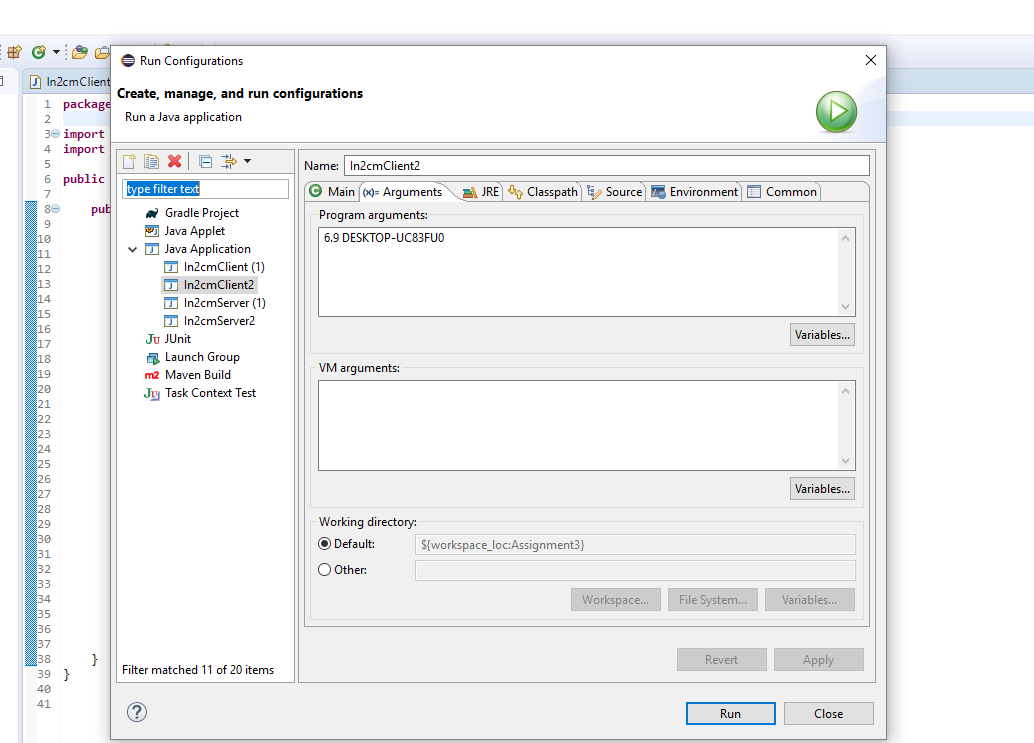
**Where the basic code was obtained and what was modified?**

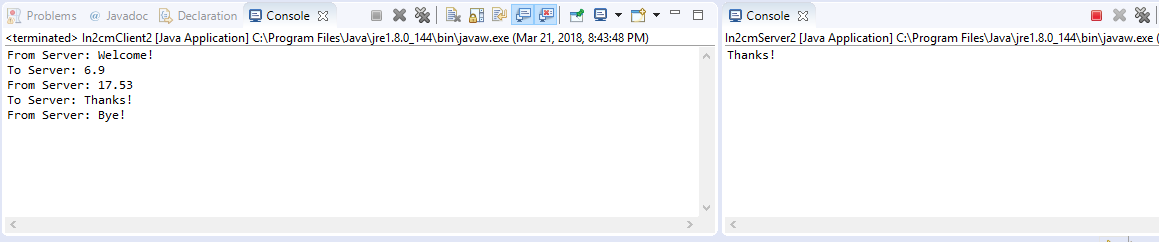
The basic code was obtained from the tutorial and was simplified (it doesn’t use buffered readers, etc). Instead of using writeBytes and readLines I used writeUTF and readUTF since it was easier to use and readLines wouldn’t work properly and as simply as readUTF did. DecimalFormat was used to send the answer from the server back to the client in the proper format (2 decimal places). Other minor improvements were added, and comments are provided in the program code.

**What is the difference between In2cmClient and In2cmClient2?**

In2cmClient2 has an extra argument which allows you to type in the host name of where the server is located (i.e on another computer). That is the only difference between the two programs.

**Screenshots**





**Source Code**

In2cmClient

**import** java.io.\*;

**import** java.net.\*;

**public** **class** In2cmClient {

**public** **static** **void** main(String argv[]) **throws** Exception {

String welcome, cm, goodbye;

**if** (argv.length != 1) {

System.***out***.println("Usage: In2cmClient [inches]");

}

Socket clientSocket = **new** Socket("host name", 24255); //change to the computers host name

DataInputStream FromClient = **new** DataInputStream(clientSocket.getInputStream());

DataOutputStream ToServer = **new** DataOutputStream(clientSocket.getOutputStream());

welcome = FromClient.readUTF(); //reads welcome from server

System.***out***.println("From Server: " + welcome);

ToServer.writeUTF(argv[0]); //sends input to server

System.***out***.println("To Server: " + argv[0]);

cm = FromClient.readUTF(); //reads answer from server

System.***out***.println("From Server: " + cm);

ToServer.writeUTF("Thanks!"); //sends thanks to server

System.***out***.println("To Server: Thanks!");

goodbye = FromClient.readUTF(); //reads bye from server

System.***out***.println("From Server: " + goodbye);

FromClient.close(); //closes client

ToServer.close(); //closes server

clientSocket.close(); //closes client socket

}

}

In2cmServer

**import** java.io.\*;

**import** java.net.\*;

**import** java.text.DecimalFormat;

**public** **class** In2cmServer {

**public** **static** **void** main( String argv[] ) **throws** Exception{

String inch, thanks;

**double** cm;

DecimalFormat df = **new** DecimalFormat("#.00"); //formats value to 2 decimal places

ServerSocket serverSocket = **new** ServerSocket(24255);

**while**(**true**) {

Socket connectionSocket = serverSocket.accept();

DataOutputStream ToClient = **new** DataOutputStream(connectionSocket.getOutputStream());

DataInputStream FromClient = **new** DataInputStream(connectionSocket.getInputStream());

ToClient.writeUTF("Welcome!"); //sends welcome to client

inch = FromClient.readUTF(); //reads client input

cm = Double.*parseDouble*(inch) \* 2.54; //converts inches to cms

ToClient.writeUTF(df.format(cm)); //sends answer to client

thanks = FromClient.readUTF(); //reads thanks from client

System.***out***.println(thanks);

ToClient.writeUTF("Bye!"); //sends bye to client

ToClient.close(); //closes client stream

connectionSocket.close(); //closes connection socket to client

}

//IMPORTANT! make sure to close server after testing is done!

}

}

In2cmClient2

**import** java.io.\*;

**import** java.net.\*;

**public** **class** In2cmClient2 {

**public** **static** **void** main(String argv[]) **throws** Exception {

String welcome, cm, goodbye;

**if** (argv.length != 2) {

System.***out***.println("Usage: In2cmClient [inches] [host name]");

}

Socket clientSocket = **new** Socket(argv[1], 24255);

DataInputStream FromClient = **new** DataInputStream(clientSocket.getInputStream());

DataOutputStream ToServer = **new** DataOutputStream(clientSocket.getOutputStream());

welcome = FromClient.readUTF(); //reads welcome from server

System.***out***.println("From Server: " + welcome);

ToServer.writeUTF(argv[0]); //sends input to server

System.***out***.println("To Server: " + argv[0]);

cm = FromClient.readUTF(); //reads answer from server

System.***out***.println("From Server: " + cm);

ToServer.writeUTF("Thanks!"); //sends thanks to server

System.***out***.println("To Server: Thanks!");

goodbye = FromClient.readUTF(); //reads bye from server

System.***out***.println("From Server: " + goodbye);

FromClient.close(); //closes client

ToServer.close(); //closes server

clientSocket.close(); //closes client socket

}

}

In2cmServer2

**import** java.io.\*;

**import** java.net.\*;

**import** java.text.DecimalFormat;

**public** **class** In2cmServer2 {

**public** **static** **void** main( String argv[] ) **throws** Exception{

String inch, thanks;

**double** cm;

DecimalFormat df = **new** DecimalFormat("#.00"); //formats value to 2 decimal places

ServerSocket serverSocket = **new** ServerSocket(24255);

**while**(**true**) {

Socket connectionSocket = serverSocket.accept();

DataOutputStream ToClient = **new** DataOutputStream(connectionSocket.getOutputStream());

DataInputStream FromClient = **new** DataInputStream(connectionSocket.getInputStream());

ToClient.writeUTF("Welcome!"); //sends welcome to client

inch = FromClient.readUTF(); //reads client input

cm = Double.*parseDouble*(inch) \* 2.54; //converts inches to cms

ToClient.writeUTF(df.format(cm)); //sends answer to client

thanks = FromClient.readUTF(); //reads thanks from client

System.***out***.println(thanks);

ToClient.writeUTF("Bye!"); //sends bye to client

ToClient.close(); //closes client stream

connectionSocket.close(); //closes connection socket to client

}

//IMPORTANT! make sure to close server after testing is done!

}

}