Name: Ken Lin

Github repository of my code : https://github.com/DehNutCase/CSE-461/tree/master/lab4

I believe all parts are completed successfully.

However, I don't know how many points they are worth (the lab 4 page doesn't say), so I assumed 20 points.

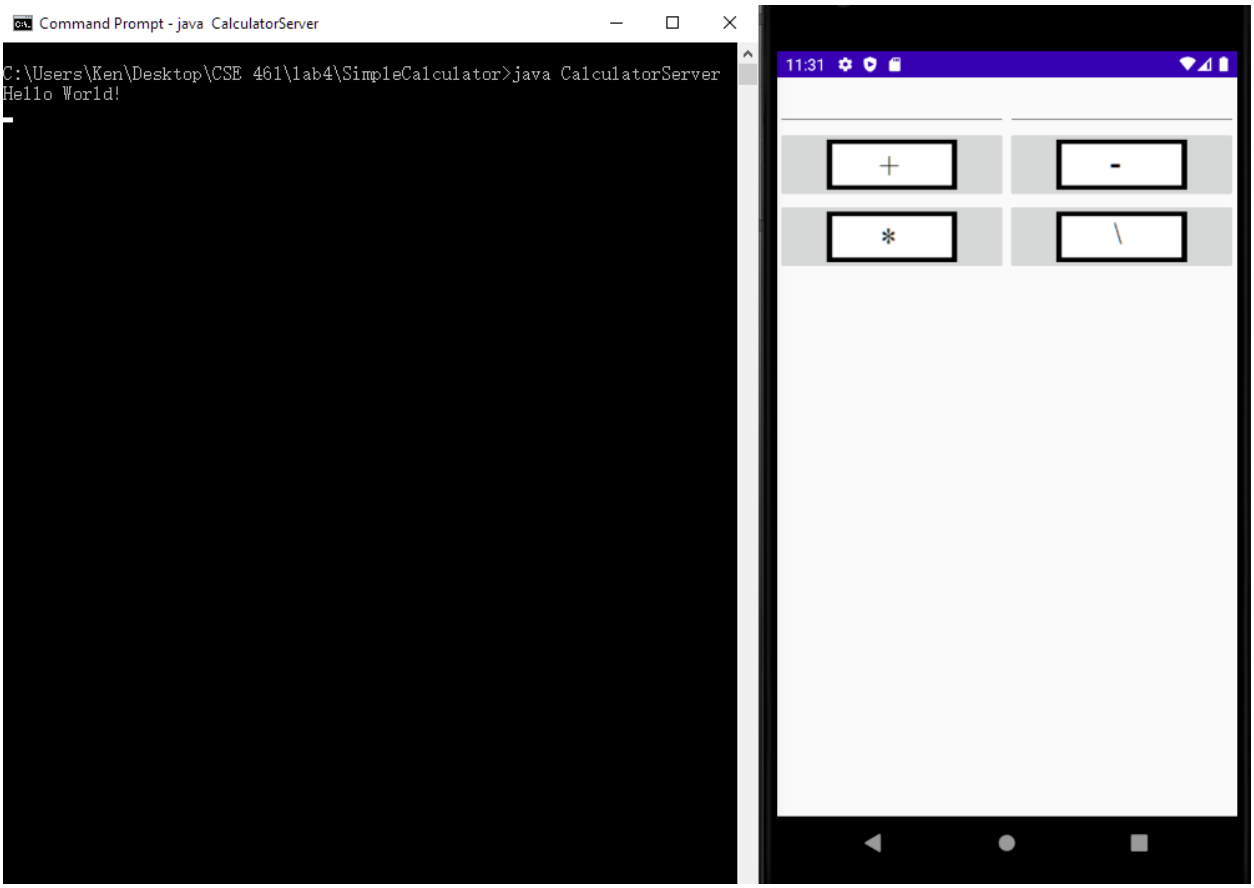
20 Points

Report:

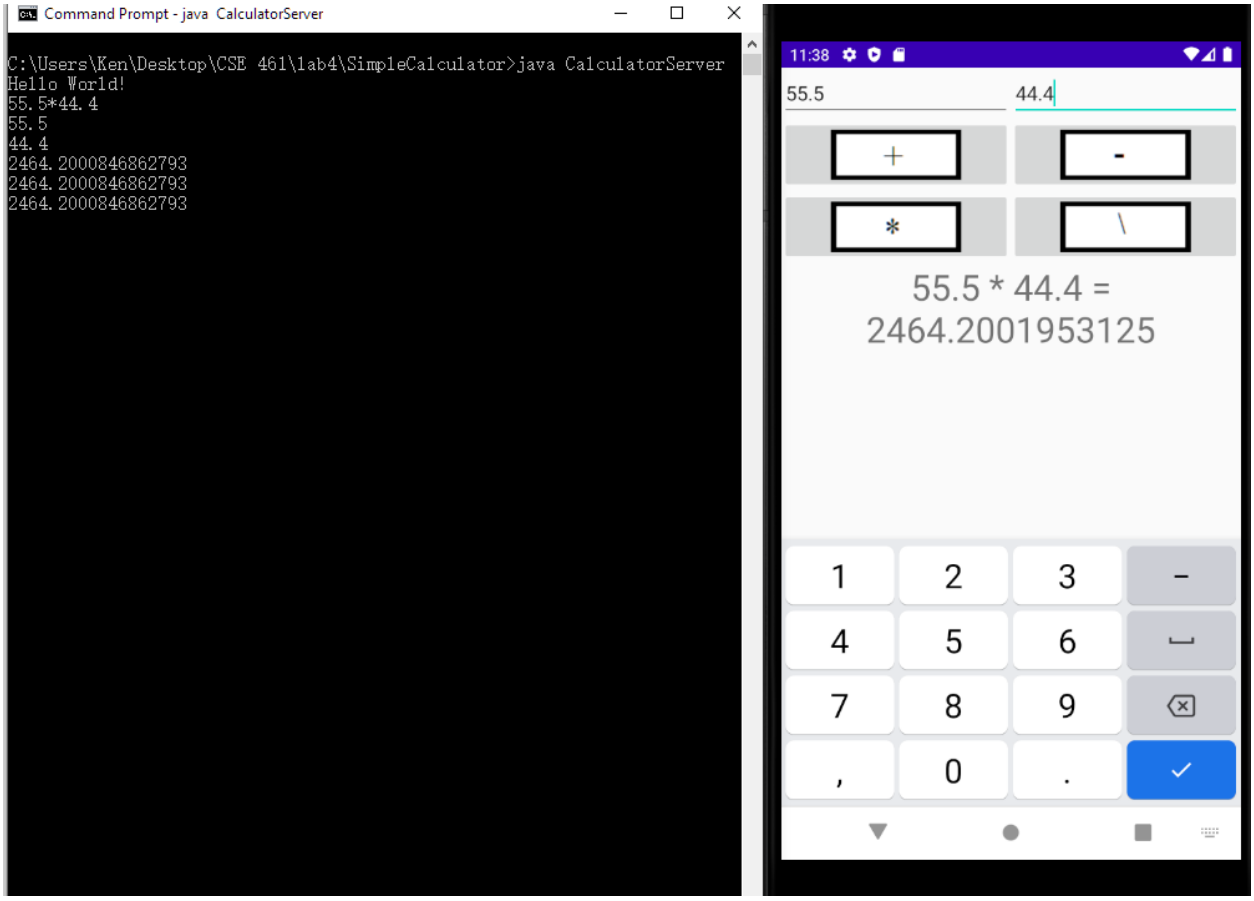
Part I, Simple Remote Calculator:

Make a calculator app with the android app being the client which sends a message to the server which actually calculates the value before sending back the results.

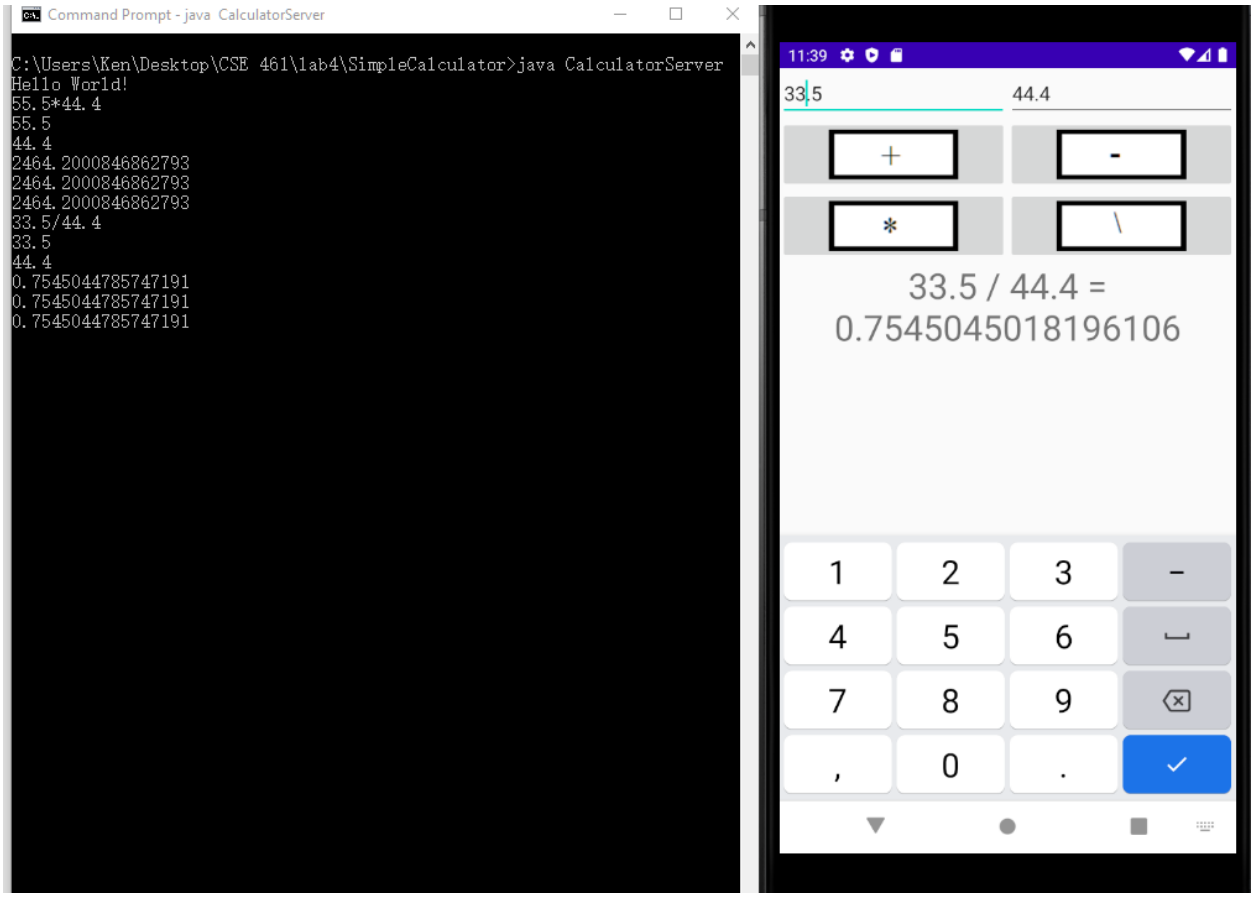
Screenshot:



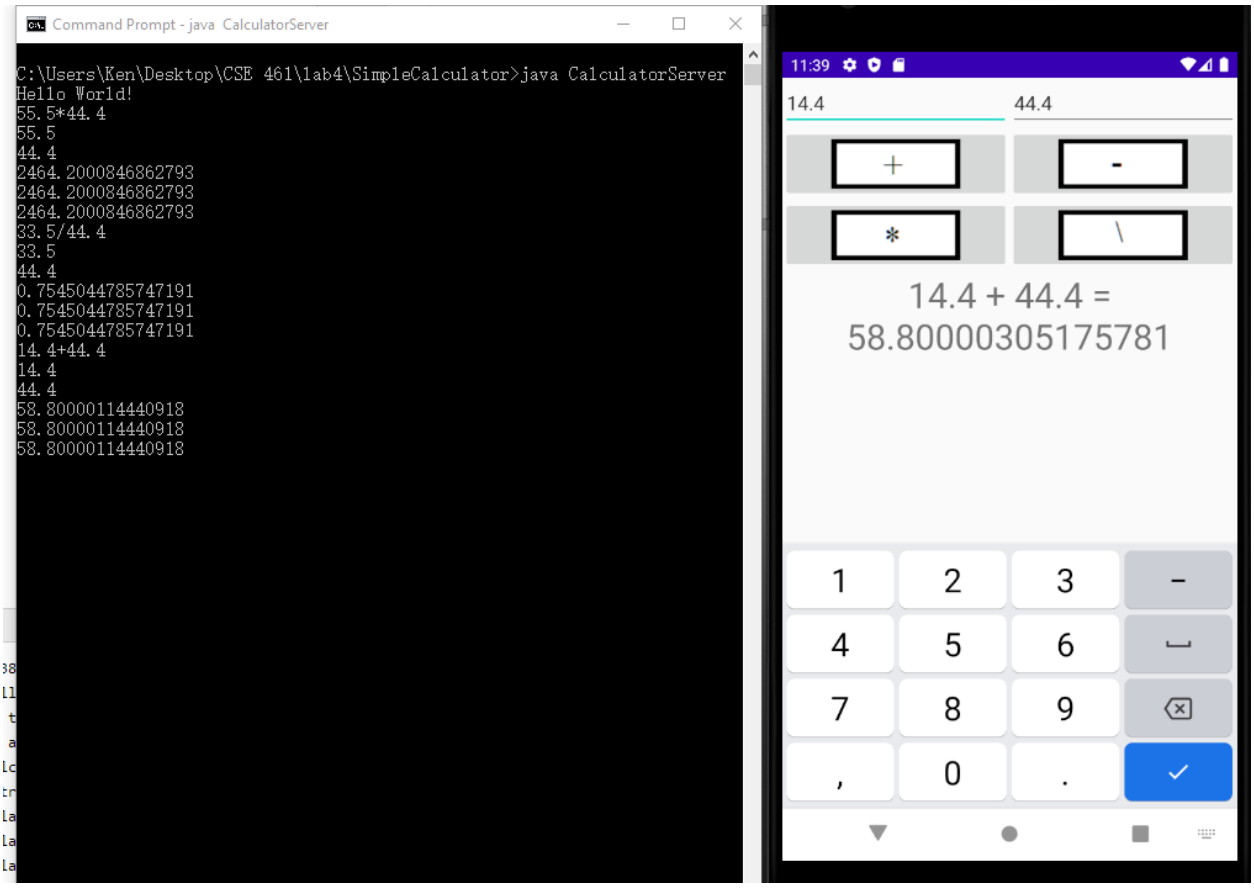
App and server started.


Command Prompt - java CalculatorServer
C:\Users\Ken\Desktop\CSE 461\lab4\SimpleCalculator>java CalculatorServer
Hello World!
55.5*44.4
55.5
44.4
2464.2000846862793
2464.2000846862793
2464.2000846862793
11:38
55.5
44.4
+
-
*
\
55.5 * 44.4 =
2464.2001953125
1
2
3
4
5
6
7
8
9
,
0
.

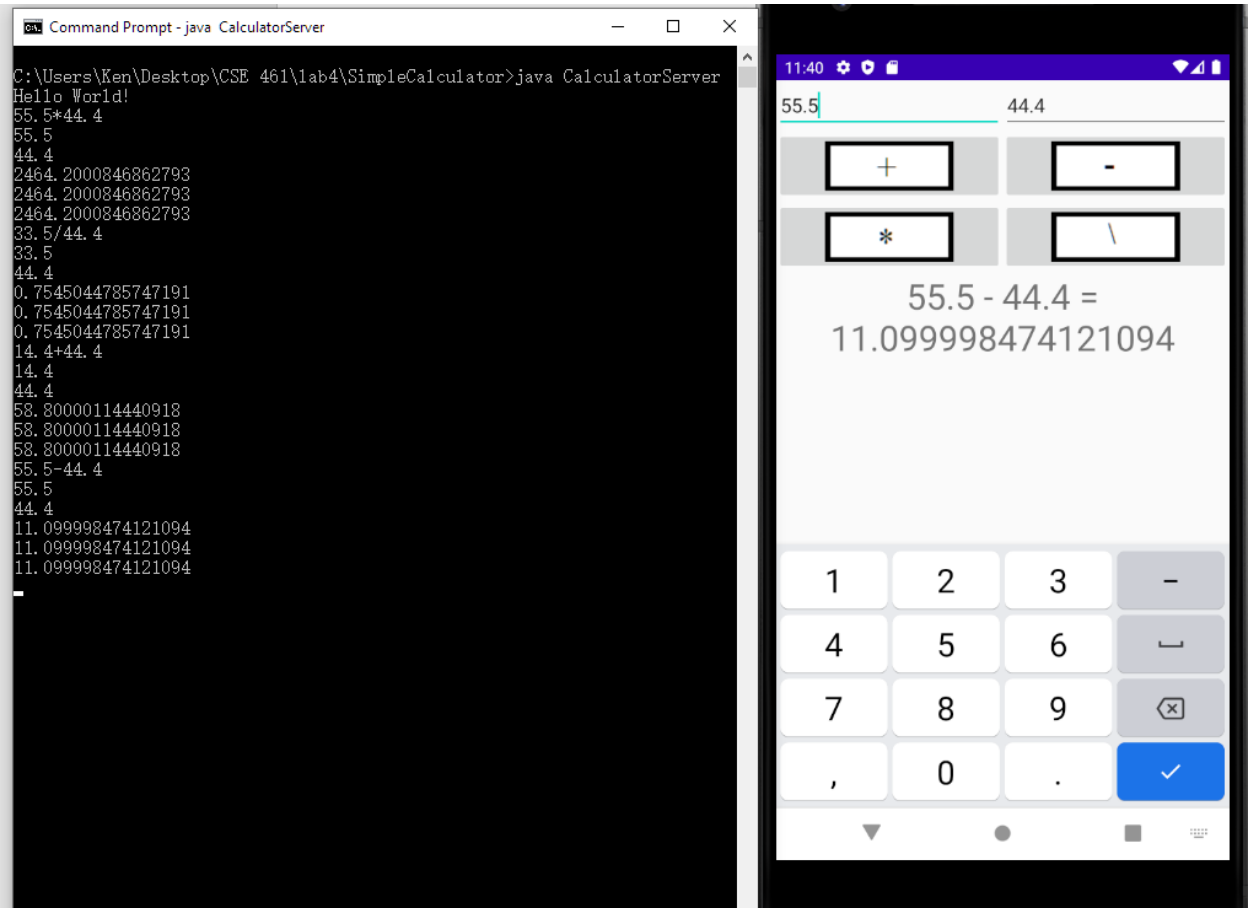
55.5 * 44.4


Command Prompt - java CalculatorServer
C:\Users\Ken\Desktop\CSE 461\lab4\SimpleCalculator>java CalculatorServer
Hello World!
55.5*44.4
55.5
44.4
2464.2000846862793
2464.2000846862793
2464.2000846862793
33.5/44.4
33.5
44.4
0.7545044785747191
0.7545044785747191
0.7545044785747191
11:39
33.5
44.4
+
-
*
\
33.5 / 44.4 =
0.7545045018196106
1
2
3
4
5
6
7
8
9
,
0
.

33.5 / 44.4



14.4 + 44.4 (trailing digits are due to rounding error from floating point.)



55.5 – 44.4

Code:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/linearLayout1"
        android:layout_marginLeft="12pt"
        android:layout_marginRight="12pt"
        android:layout_marginTop="4pt">
        <EditText
            android:layout_weight="1"
            android:layout_height="wrap_content"
            android:layout_marginRight="6pt"
            android:id="@+id/t1"
            android:layout_width="match_parent"
```

```
            android:inputType="numberDecimal">
        </EditText>
        <EditText
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:layout_marginLeft="6pt"
            android:id="@+id/t2"
            android:layout_width="match_parent"
            android:inputType="numberDecimal">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/linearLayout2"
        android:layout_marginTop="4pt"
        android:layout_marginLeft="6pt"
        android:layout_marginRight="6pt">
        <ImageButton
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:layout_weight="1"
            android:src="@drawable/plus_button"
            android:id="@+id/plus">
        </ImageButton>
        <ImageButton
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:layout_weight="1"
            android:src="@drawable/minux_button"
            android:id="@+id/minus">
        </ImageButton>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/linearLayout3"
        android:layout_marginTop="4pt"
        android:layout_marginLeft="6pt"
        android:layout_marginRight="6pt">

        <ImageButton
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:layout_weight="1"
            android:src="@drawable/multiply_button"
            android:id="@+id/multiply">
        </ImageButton>
        <ImageButton
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:layout_weight="1"
            android:src="@drawable/divide_button"
            android:id="@+id/divide">
```

```
        </ImageButton>
    </LinearLayout>

    <TextView
        android:id="@+id/displayResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:gravity="center_horizontal"
        android:textSize="32dp">
    </TextView>
</LinearLayout>
```

strings.xml

```
<resources>
    <string name="app_name">Lab3SimpleCalculator</string>
</resources>
```

MainActivity.java (This is called lab3simplecalculator since I initialized the project by copying the lab3 code.)

```
package com.example.lab3simplecalculator;

import androidx.appcompat.app.AppCompatActivity;
//import android.support.v7.app.AppCompatActivity;
import android.content.Context;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.os.StrictMode;
import android.text.TextUtils;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.EditText;
import android.widget.TextView;
import android.app.Activity;
import android.app.ActionBar;
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
```

```
import android.os.Build;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class MainActivity extends Activity implements View.OnClickListener{
    EditText t1;
    EditText t2;

    ImageButton plus;
    ImageButton minus;
    ImageButton multiply;
    ImageButton divide;

    TextView displayResult;

    String oper = "";

    private Socket client_socket;
    private static final int SERVERPORT = 5665; //makesure this matches the port in
CalculatorServer.java
    private static final String SERVER_IP = "192.168.0.100"; //ipconfig gets this

    String num1 = "0";
    String num2 = "0";

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // find the EditText elements (defined in res/layout/activity_main.xml
        t1 = (EditText) findViewById(R.id.t1);
        t2 = (EditText) findViewById(R.id.t2);

        plus = (ImageButton) findViewById(R.id.plus);
        minus = (ImageButton) findViewById(R.id.minus);
        multiply = (ImageButton) findViewById(R.id.multiply);
        divide = (ImageButton) findViewById(R.id.divide);

        displayResult = (TextView) findViewById(R.id.displayResult);

        // set  listeners
        plus.setOnClickListener(this);
        minus.setOnClickListener(this);
        multiply.setOnClickListener(this);
        divide.setOnClickListener(this);
```

```
        new Thread(new ClientThread()).start();
    }

    PrintWriter out;
    BufferedReader input;
    class ClientThread implements Runnable {
        @Override
        public void run() {
            try {
                InetAddress serverAddr = InetAddress.getByName(SERVER_IP);
                client_socket = new Socket(serverAddr, SERVERPORT);
                out = new PrintWriter(new BufferedWriter(
                        new OutputStreamWriter(client_socket.getOutputStream())),
                        true);

                input = new BufferedReader(new
InputStreamReader(client_socket.getInputStream()));
            } catch (UnknownHostException e1) {
                e1.printStackTrace();
            } catch (IOException e1) {
                e1.printStackTrace();
            }
        }
    }

    // @Override
    public void onClick( View view ) {



        // check if the fields are empty
        if (TextUtils.isEmpty(t1.getText().toString())
                || TextUtils.isEmpty(t2.getText().toString())) {
            return;
        }

        // read EditText and fill variables with numbers
        num1 = t1.getText().toString();
        num2 = t2.getText().toString();

        String str = "";

        // perform operations
        // save operator in oper for later use
        switch ( view.getId() ) {
            case R.id.plus:
                oper = "+";
                str = num1 + oper + num2;
                break;
            case R.id.minus:
                oper = "-";
                str = num1 + oper + num2;
                break;
            case R.id.multiply:
```

```
                oper = "*";
                str = num1 + oper + num2;
                break;
            case R.id.divide:
                oper = "/";
                str = num1 + oper + num2;
                break;
            default:
                break;
        }

        String[] str_list = {str};

        SendfeedbackJob job = new SendfeedbackJob();
        job.execute(str);
        str = "";


    }

    private class SendfeedbackJob extends AsyncTask<String, Void, String> {
        double result = 0;
        @Override
        protected String doInBackground(String[] params) {
            try {
                out.println(params[0]);
                String inputLine = null;
                int index = 0;
                while ( ( inputLine = input.readLine() )  != null ) {
                    result = Float.parseFloat(inputLine);
                    index = index + 1;
                    return "Done!";
                }
            } catch (UnknownHostException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            } catch (Exception e) {
                e.printStackTrace();
            }
            return "Message sent to server and result received.";
        }

        @Override
        protected void onPostExecute(String message) {
            //process message
            // form the output line
            displayResult.setText(num1 + " " + oper + " " + num2 + " = " + result);
        }
    }
}
```

AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.lab3simplecalculator">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:usesCleartextTraffic="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
</manifest>
```

CalculatorServer.java

```
// A simple TCP server for Demo
// @Author: T.L. Yu

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket;

import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.io.BufferedWriter;

public class CalculatorServer {
  public static void main(String[] args) throws IOException {

    if (args.length != 0) {
      System.err.println("Usage: java CalculatorServer");
    }
```

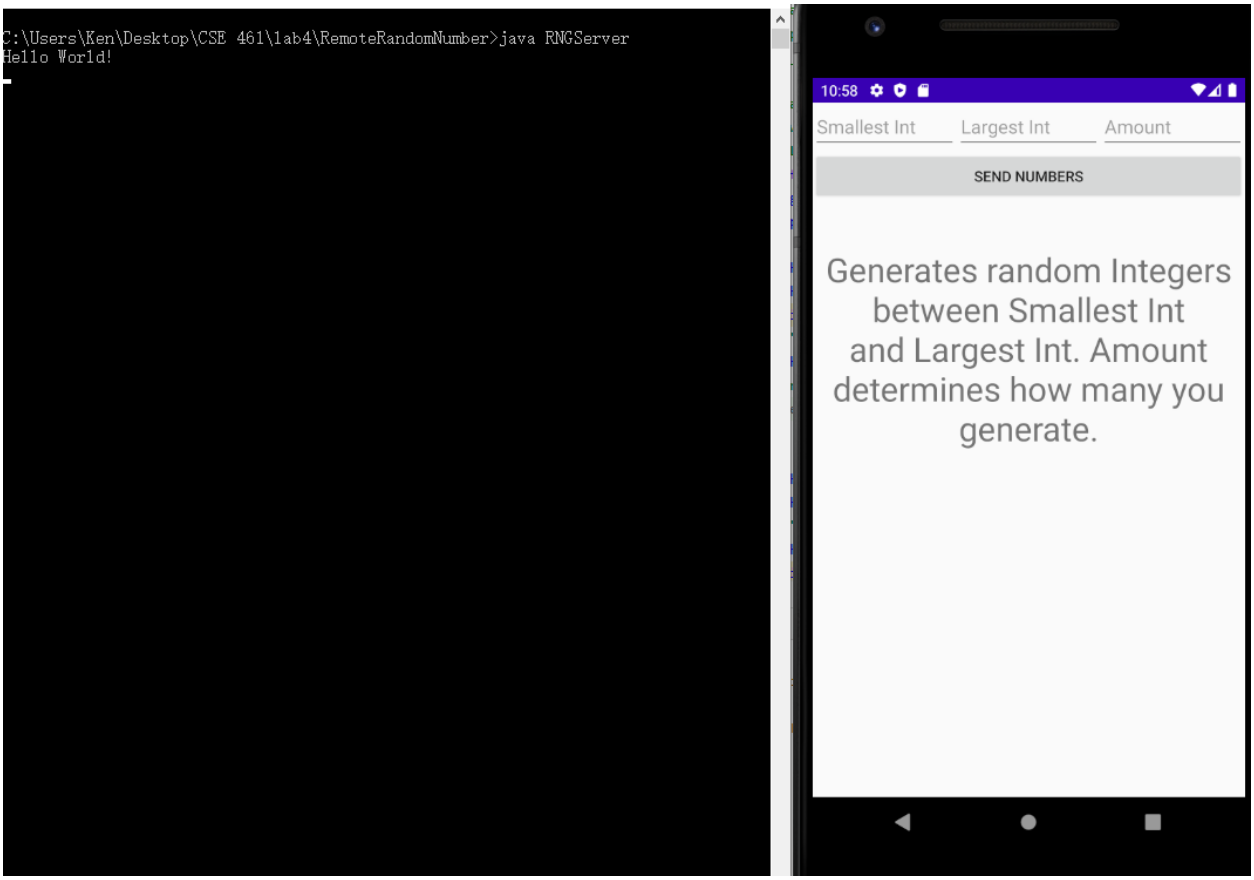
```
int portNumber = 5665;
System.out.println("Hello World!");

try {
  ServerSocket serverSocket =  new ServerSocket(portNumber);
  Socket clientSocket = serverSocket.accept();
  BufferedReader input = new BufferedReader (
       new InputStreamReader(clientSocket.getInputStream()));

   String inputLine = null;
   while ( ( inputLine = input.readLine() )  != null ) {
    //print input line for debugging purposes
    System.out.println ( inputLine );

    //parse string and calculate result
    String string_array[] = inputLine.split("[+\\-*/]");
    System.out.println (string_array[0]);
    System.out.println (string_array[1]);
    double num1 = Float.parseFloat(string_array[0]);
    double num2 = Float.parseFloat(string_array[1]);
    String oper = "+";
    int index = 0;
    double result = 0;
    while(true){
      boolean found = false;
      switch(inputLine.charAt(index)){
        case '+':
          result = num1 + num2;
          found = true;
          break;
        case '-':
          result = num1 - num2;
          found = true;
          break;
        case '*':
          result = num1 * num2;
          found = true;
          break;
        case '/':
          result = num1/num2;
          found = true;
          break;
      }
```

```
        if (found) break;
        index = index + 1;
       }
       System.out.println(result);
       //send result to client
       PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
                    true);
       System.out.println(result);
       out.println(result);
       System.out.println(result);
      }
    }  catch (IOException e) {
          System.out.println("Exception caught when trying to listen on port "
             + portNumber + " or listening for a connection");
          System.out.println(e.getMessage());
    }
  }
}
```

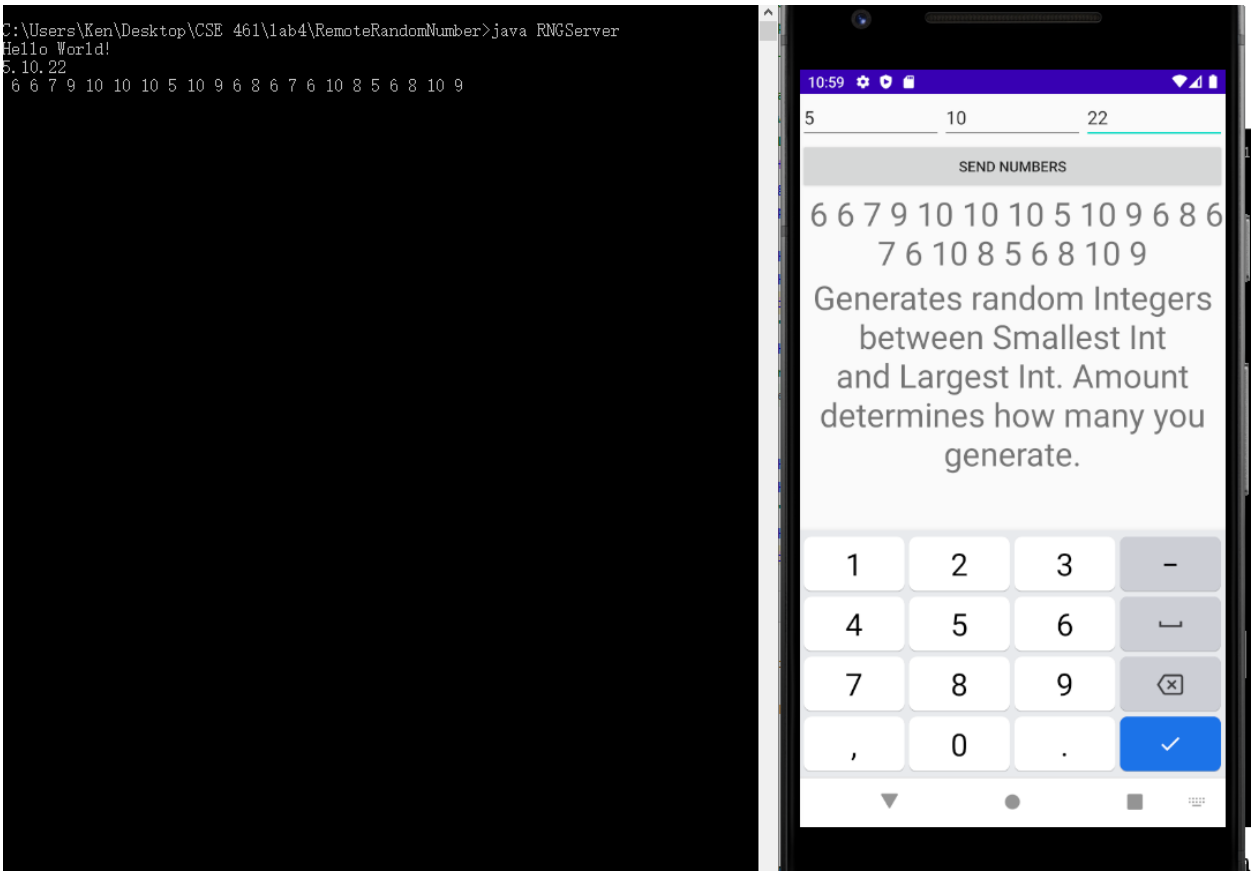
Part II, Remote Random Number Generator:

Android client app that requests random numbers from a server.

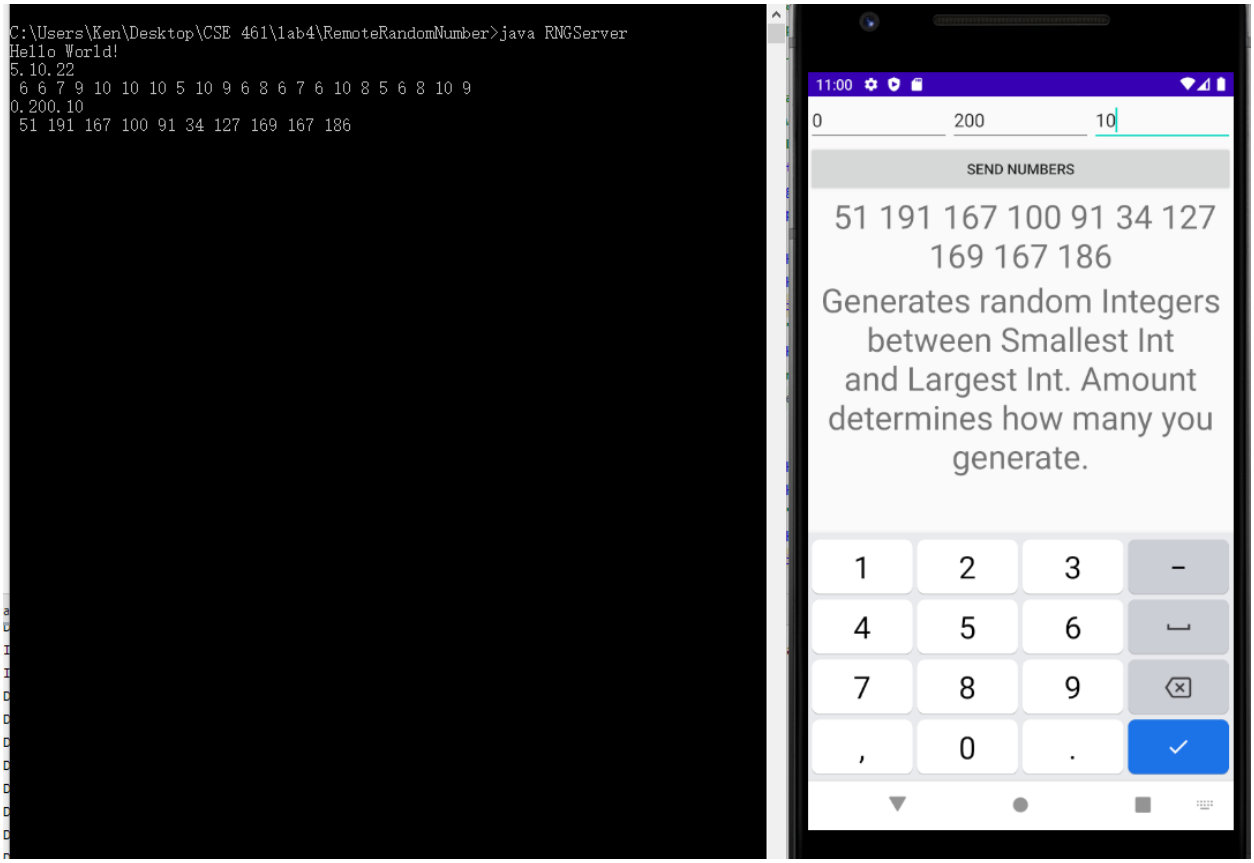
Screenshot:



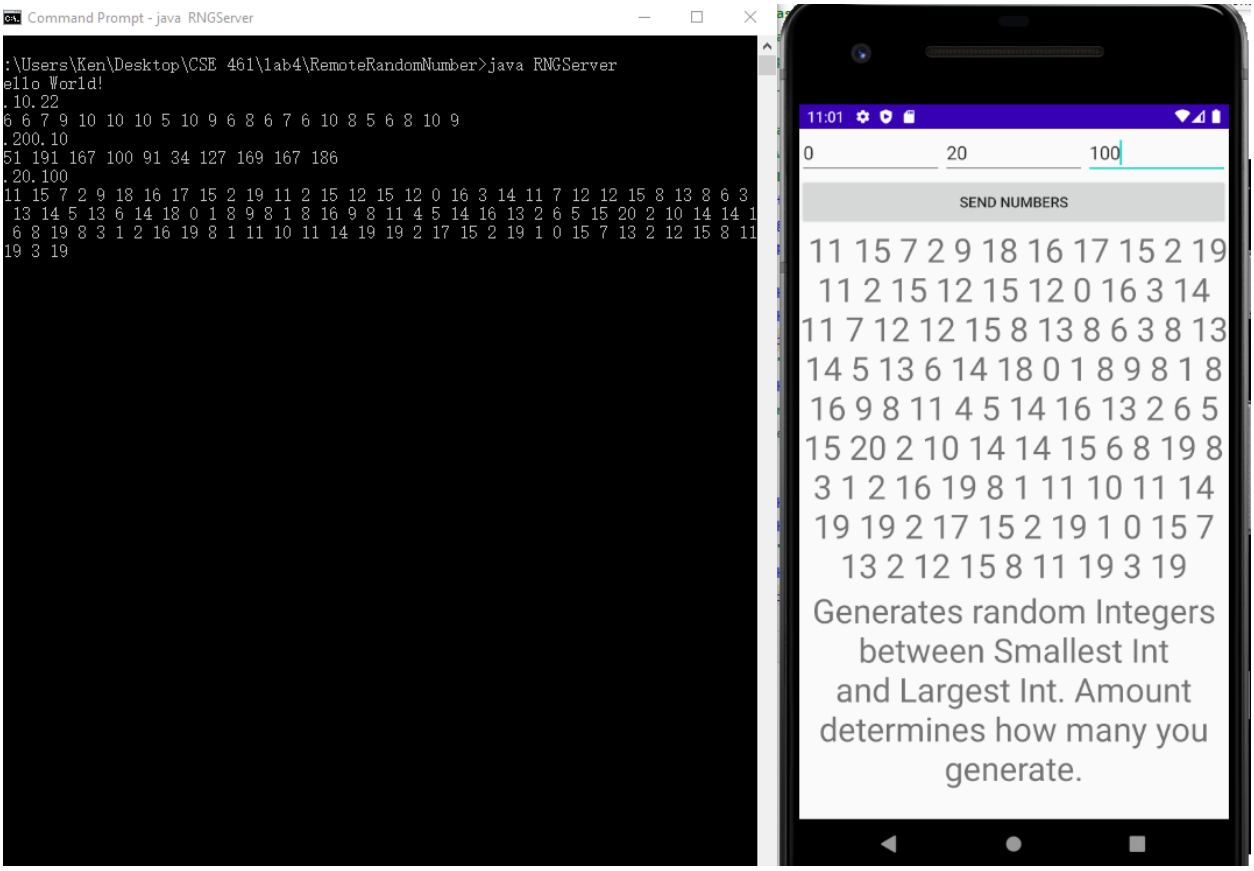
Initial start of up with server online.



Requesting 22 numbers between 5 and 10.



Requesting 10 numbers between 0 and 200.



100 numbers between 0 and 20.

Code:

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/linearLayout1"
        android:layout_marginLeft="12pt"
        android:layout_marginRight="12pt"
        android:layout_marginTop="4pt">
        <EditText
            android:layout_weight="1"
            android:layout_height="wrap_content"
            android:layout_marginRight="6pt"
            android:id="@+id/t1"
            android:layout_width="match_parent"
```

```
            android:inputType="numberDecimal"
            android:hint="@string/smallest_int">
        </EditText>
        <EditText
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:id="@+id/t2"
            android:layout_width="match_parent"
            android:layout_marginRight="6pt"
            android:inputType="numberDecimal"
            android:hint="@string/largest_int">
        </EditText>
        <EditText
            android:layout_weight="1"
            android:layout_height="wrap_content"
            android:id="@+id/t3"
            android:layout_width="match_parent"
            android:inputType="numberDecimal"
            android:hint="@string/amount">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/linearLayout2"
        android:layout_marginTop="4pt"
        android:layout_marginLeft="6pt"
        android:layout_marginRight="6pt">
        <Button
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:layout_weight="1"
            android:text="@string/send_numbers"
            android:id="@+id/send_button">
        </Button>
    </LinearLayout>

    <TextView
        android:id="@+id/displayResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:gravity="center_horizontal"
        android:textSize="32dp"
        >
    </TextView>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/linearLayout3"
        android:layout_marginTop="4pt"
        android:layout_marginLeft="6pt"
        android:layout_marginRight="6pt">
```

```
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:gravity="center_horizontal"
            android:text="@string/Instruction_Text"
            android:textSize="32dp">
        </TextView>
    </LinearLayout>
</LinearLayout>
```

strings.xml

```
<resources>
    <string name="app_name">Lab4RemoteRandomNumber</string>
    <string name="Instruction_Text">Generates random Integers between Smallest Int
and Largest Int. Amount determines how many you generate.</string>
    <string name="send_numbers">Send Numbers</string>
    <string name="amount">Amount</string>
    <string name="largest_int">Largest Int</string>
    <string name="smallest_int">Smallest Int</string>
</resources>
```

MainActivity.java

```
package com.example.lab4remoterandomnumber;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.os.StrictMode;
import android.text.TextUtils;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.EditText;
import android.widget.TextView;
import android.app.Activity;
import android.app.ActionBar;
import android.app.Fragment;
import android.os.Bundle;
```

```
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.os.Build;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class MainActivity extends Activity implements View.OnClickListener{
    EditText t1;
    EditText t2;
    EditText t3;

    Button send;

    TextView displayResult;

    String oper = "";

    private Socket client_socket;
    private static final int SERVERPORT = 6556; //makesure this matches the port in
RNGServer.java
    private static final String SERVER_IP = "192.168.0.100"; //ipconfig gets this

    String num1 = "0";
    String num2 = "0";
    String num3 = "1";

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // find the EditText elements (defined in res/layout/activity_main.xml
        t1 = (EditText) findViewById(R.id.t1);
        t2 = (EditText) findViewById(R.id.t2);
        t3 = (EditText) findViewById(R.id.t3);

        send = (Button) findViewById(R.id.send_button);

        displayResult = (TextView) findViewById(R.id.displayResult);

        // set  listeners
        send.setOnClickListener(this);

        new Thread(new ClientThread()).start();
```

```
    }

    PrintWriter out;
    BufferedReader input;
    class ClientThread implements Runnable {
        @Override
        public void run() {
            try {
                InetAddress serverAddr = InetAddress.getByName(SERVER_IP);
                client_socket = new Socket(serverAddr, SERVERPORT);
                out = new PrintWriter(new BufferedWriter(
                        new OutputStreamWriter(client_socket.getOutputStream())),
                        true);

                input = new BufferedReader(new
InputStreamReader(client_socket.getInputStream()));
            } catch (UnknownHostException e1) {
                e1.printStackTrace();
            } catch (IOException e1) {
                e1.printStackTrace();
            }
        }
    }

    // @Override
    public void onClick( View view ) {

        // check if the fields are empty
        if (TextUtils.isEmpty(t1.getText().toString())
                || TextUtils.isEmpty(t2.getText().toString())
                || TextUtils.isEmpty(t3.getText().toString())) {
            return;
        }

        // read EditText and fill variables with numbers
        num1 = Integer.toString(Integer.parseInt(t1.getText().toString()));
//forcibly checks that num 1 is an int rather than a float
        num2 = Integer.toString(Integer.parseInt(t2.getText().toString()));
        num3 = Integer.toString(Integer.parseInt(t3.getText().toString()));

        String str = num1 + "." + num2 + "." + num3;

        SendfeedbackJob job = new SendfeedbackJob();
        job.execute(str);
        str = "";
    }

    private class SendfeedbackJob extends AsyncTask<String, Void, String> {
        String result = "";
        @Override
        protected String doInBackground(String[] params) {
            try {
                out.println(params[0]);
                String inputLine = null;
                int index = 0;
```

```
                while ( ( inputLine = input.readLine() )  != null ) {
                    result = inputLine;
                    index = index + 1;
                    return "Done!"; //doesn't work without this line, I don't think
I'm using Asynch Task properly
                }
            } catch (UnknownHostException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            } catch (Exception e) {
                e.printStackTrace();
            }
            return "Message sent to server and result received.";
        }

        @Override
        protected void onPostExecute(String message) {
            //process message
            // form the output line
            displayResult.setText(result);
        }
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.lab4remoterandomnumber">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:usesCleartextTraffic="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
</manifest>
```

RNGServer.java (I kept the comments from the sample server file I got from Dr. Yu)

```
// A simple TCP server for Demo
// @Author: T.L. Yu

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket;

import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.io.BufferedWriter;

import java.util.Random;

public class RNGServer {
  public static void main(String[] args) throws IOException {

    if (args.length != 0) {
      System.err.println("Usage: java RNGServer");
    }

    int portNumber = 6556;
    System.out.println("Hello World!");

    try {
      ServerSocket serverSocket =  new ServerSocket(portNumber);
      Socket clientSocket = serverSocket.accept();
      BufferedReader input = new BufferedReader (
          new InputStreamReader(clientSocket.getInputStream()));

       String inputLine = null;
       while ( ( inputLine = input.readLine() )  != null ) {
         //print input line for debugging purposes
        System.out.println ( inputLine );

        //parse string and calculate result
        String string_array[] = inputLine.split("\\D");
        int num1 = Integer.parseInt(string_array[0]);
        int num2 = Integer.parseInt(string_array[1]);
```

```
        int num3 = Integer.parseInt(string_array[2]);

        int index = 0;
        String result = "";
        StringBuilder result_builder = new StringBuilder();

        if(num2 < num1){
          //make sure num1 is <= num2
          int temp = num1;
          num1 = num2;
          num2 = temp;
        }

        Random rand = new Random();

        while (index < num3){
          index = index + 1;

          //generates a random integer between num1 and num2
          int new_random = rand.nextInt(num2 - num1 + 1);
          new_random = new_random + num1;

          if (index != 0){
            result_builder.append(" "); //make sure numbers are separated by space
          }
          result_builder.append(new_random);  //add number to string
        }

        result = result_builder.toString();

        //send result to client
        PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
                        true);
        out.println(result);
        System.out.println(result); //printing result to be sure it worked
       }
     }  catch (IOException e) {
           System.out.println("Exception caught when trying to listen on port "
              + portNumber + " or listening for a connection");
           System.out.println(e.getMessage());
     }
   }
}
```