

# **Q3 Code Writeup**

**Luke Knepper**

### ***Continuous Authentication Simulation Program:***

**DriverProgram.java:** *This is the program which runs the simulation of the continuous authentication procedure. The program simulates an instant-messaging application, as it has a large text area where messages are displayed and a smaller text area for text entry. There is a simple script which simulates a person talking to the user and asking the user questions, which the user is expected to respond to. The first question asks the user to type a simple sentence which comprises all of the letters of the alphabet, and the program measures the user's typing characteristics and uses them to train a neural network (one of the classes below which implement NN.java). Then the program continues to ask the user random questions, and measures the user's typing characteristics while the user is typing responses. After each response, the program sends the user's typing data through the neural network, which outputs a value. If the outputted value is too low, i.e. lower than the value outputted by the typing data of the first sentence the user typed, then that signals that the user's typing characteristics may have changed, which means that there may be a different user using the account, and the warning level is raised (currently by 10%). If the value meets the threshold, i.e. exceeds the value outputted by the typing data of the first sentence the user typed, then the program does not detect a problem, and slightly lowers the warning level (currently by 1%). If the warning level reaches a critical value (currently 100%), that signals that an intruder may have gained access to the user's account, and the system locks the user out for security.*

```
// Dynamic Authentication by Typing Patterns
```

```
// Main Program Code
```

```
// Luke Knepper, 2009
```

```
// luke@lukeknepper.com
```

```
// *All rights reserved.*
```

```
import java.awt.BorderLayout;
```

```
import java.awt.Container;
```

```
import java.awt.Dimension;
```

```
import java.awt.event.*;
```

```
import javax.swing.*;
```

```
import java.util.*;
```

```
import java.util.Random;
```

```
import java.awt.Font;
```

```
import java.lang.*;
```

```
class MessageThread implements Runnable {
```

```
    Thread runner;
```

```
    JTextArea displayArea;
```

```
    String s;
```

```
    public MessageThread(JTextArea displayArea_t, String s_t) {
```

```
        displayArea = displayArea_t;
```

```
        s = s_t;
```

```
    }
```

```
    public MessageThread(String threadName, JTextArea displayArea_t, String s_t) {
```

```
        displayArea = displayArea_t;
```

```
        s = s_t;
```

```
        runner = new Thread(this, threadName); // (1) Create a new thread.
```

```
//        System.out.println(runner.getName());
```

```
        runner.start(); // (2) Start the thread.
```

```

    }
    public void run() {
        //Display info about this particular thread
        System.out.println(Thread.currentThread());
        try {
            Thread.sleep(300);
        } catch(InterruptedException e1) {}
        displayMessage(" *Leila is typing... *");
        try {
            Thread.sleep(1000);
        } catch(InterruptedException e1) {}
        displayMessage(s);
    }
    private void displayMessage(String t){
        displayArea.append(t + "\n");
        displayArea.setCaretPosition( displayArea.getText().length() );
    }
}

```

```

public class DriverProgram extends JFrame
    implements KeyListener,
    ActionListener
{

```

```

    NNSingleLayer network = new NNSingleLayer();
    boolean t2 = true;
    int warning = 0;
    JTextArea displayArea;
    JTextField typingArea;
    static final String newline = System.getProperty("line.separator");
    //Format is KeyCode, data
    static Map<Integer, ArrayList<Long>> charMap1 = new HashMap<Integer, ArrayList<Long>>();
    static Map<Integer, ArrayList<Long>> charMap2 = new HashMap<Integer, ArrayList<Long>>();
    static Map<Integer, ArrayList<Long>> charMap3 = new HashMap<Integer, ArrayList<Long>>();
    Map<Integer, Long> curKeyTimes = new HashMap<Integer, Long>();
    //String typeText = "the quick brown fox jumps over the lazy brown dog on a saturday afternoon in
brooklyn near the bay bridge without coughing or stopping at the crosswalk and then takes a bite of pizza
without paying any money for it and then goes home and sleeps <ENTER>";
    String typeText = "The quick brown fox jumps over the lazy dog.";
    String name = "";
    int stage = 0;

    Random rand;
    Font boldFont = new Font("Serif", Font.BOLD, 12);

```

```

    String[] questions = {"what's your favorite flavor of ice cream?","where do you live?","ask me a
question!","describe your night time routine.","describe your dream house.",
    "do you prefer the beach or the mountains? Why?","if you could have an superpower, which would you

```

have?","what is/was your favorite subject in school?",

"what kind of music do you like to listen to?","if you were going to be stranded on an island for a year and could only bring three things, what would you bring?",

"how much wood would a woodchuck chuck if a woodchuck could chuck wood?","how many pickled peppers did Peter Piper pick?",

"are you creeped out by this automated question asker thing?","if you could have lunch with anyone in history, with whom would you dine?",

"have we met before?","if you could only eat one food for the rest of your life, what would you eat?","what would you do for a million dollars?",

"what's the coolest thing you've ever done?","if you could live anywhere, where would you live?","would you rather have twelve kids or no kids at all?",

"what would you want to have for your last meal on earth?","if you could go to any planet, where would you go?","what's the best invention that anyone has ever invented?",

"how often do you dream?","describe your personality."};

String[] quest\_starts = {"So, tell me,","While we're talking,","I've got a question,","I've been thinking,","On another note,","Speaking of that,"},

"I'm getting bored,","I'm wondering,","Help me get to know you better;","So,","And","Why don't you tell me,","And now for something completely different!",

"While I've got you online,","Since we're on that subject,"};

String[] answers = {"Hmm... interesting. ","That sounds cool!","Bleh, that's boring...","I like that!","Oh, you're one of THOSE people...","Haha, you're funny."},

"Bum bum bum!","Oh you...","Hah, that's silly!","Never met anyone like you before!","You sound familiar...","Like my brother!","Whoah!","Yowzers!",

"I've been waiting my whole life for someone to say that!","I forgot what I was gonna say...","Oh no! Brain overload!","I'm gonna forget that in about five minutes...",

"You're quite a memorable character. ","I disagree. ","That's just like me!","No way!","For real? That's cool. ","You lie!","Get out!","Oh, you're no fun anymore..."};

```
public static void main(String[] args) {
    /* Use an appropriate Look and Feel */
    try {
        UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
    } catch (UnsupportedLookAndFeelException ex) {
        ex.printStackTrace();
    } catch (IllegalAccessException ex) {
        ex.printStackTrace();
    } catch (InstantiationException ex) {
        ex.printStackTrace();
    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    }
    /* Turn off metal's use of bold fonts */
    UIManager.put("swing.boldMetal", Boolean.FALSE);

    //Schedule a job for event dispatch thread:
    //creating and showing this application's GUI.
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
```

```

        createAndShowGUI());
    }
});
}

/**
 * Create the GUI and show it. For thread safety,
 * this method should be invoked from the
 * event-dispatching thread.
 */
private static void createAndShowGUI() {
    //Create and set up the window.
    DriverProgram frame = new DriverProgram("Typing Security Simulation");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Set up the content pane.
    frame.addComponentsToPane();

    //Display the window.
    frame.pack();
    frame.setVisible(true);
}

private void addComponentsToPane() {

    JButton button = new JButton("Clear");
    button.addActionListener(this);

    typingArea = new JTextField(20);
    typingArea.addKeyListener(this);

    //Uncomment this if you wish to turn off focus
    //traversal. The focus subsystem consumes
    //focus traversal keys, such as Tab and Shift Tab.
    //If you uncomment the following line of code, this
    //disables focus traversal and the Tab events will
    //become available to the key event listener.
    //typingArea.setFocusTraversalKeysEnabled(false);

    displayArea = new JTextArea();
    displayArea.setLineWrap(true);
    displayArea.setWrapStyleWord(true);
    displayArea.setEditable(false);
    JScrollPane scrollPane = new JScrollPane(displayArea);
    scrollPane.setPreferredSize(new Dimension(375, 125));

    getContentPane().add(scrollPane, BorderLayout.CENTER);
    getContentPane().add(typingArea, BorderLayout.PAGE_END);
    //getContentPane().add(button, BorderLayout.PAGE_END);
}

```

```
    displayMessage(" * Leila: Hey, I'm Leila, the automated computer response system! Today we're going  
to do a little experiment with your typing patterns."+
```

```
    "\n * Leila: So, what's your name?");
```

```
    }
```

```
public DriverProgram(String name) {  
    super(name);
```

```
}
```

```
/** Handle the key typed event from the text field. */
```

```
public void keyTyped(KeyEvent e) {
```

```
    //displayInfo(e, "KEY TYPED: ");
```

```
}
```

```
/** Handle the key pressed event from the text field. */
```

```
public void keyPressed(KeyEvent e) {
```

```
    //displayInfo(e, "KEY PRESSED: ");
```

```
    //add key info and time to the curKeyTimes map
```

```
    curKeyTimes.put(e.getKeyCode(), System.currentTimeMillis());
```

```
    //if(curKeyTimes.containsKey(e.getKeyCode()) {
```

```
        //    curKeyTimes.
```

```
}
```

```
public String getr(String[] s) {
```

```
    rand = new Random();
```

```
    return s[rand.nextInt(s.length)];
```

```
}
```

```
/** Handle the key released event from the text field. */
```

```
public void keyReleased(KeyEvent e) {
```

```
    //displayInfo(e, "KEY RELEASED: ");
```

```
    //take key info from curKeyTimes map and put into charMap
```

```
    long val = System.currentTimeMillis() - curKeyTimes.get(e.getKeyCode());
```

```
    if(stage == 1) {
```

```
        if(charMap1.containsKey(e.getKeyCode()))
```

```
            charMap1.get(e.getKeyCode()).add(val);
```

```
        else {
```

```
            ArrayList<Long> temp = new ArrayList<Long>();
```

```
            temp.add(val);
```

```
            charMap1.put(e.getKeyCode(), temp);
```

```
        }
```

```
    }
```

```
    if(stage > 1) {
```

```

        if(charMap3.containsKey(e.getKeyCode()))
            charMap3.get(e.getKeyCode()).add(val);
        else {
            ArrayList<Long> temp = new ArrayList<Long>();
            temp.add(val);
            charMap3.put(e.getKeyCode(),temp);
        }
    }
    /*
    if(stage <=3 ) {
        if((stage == 1 ? charMap1 : (stage == 2?charMap2:charMap3)).containsKey(e.getKeyCode()))
            (stage == 1 ? charMap1 : (stage == 2?
charMap2:charMap3)).get(e.getKeyCode()).add(val);
        else {
            ArrayList<Long> temp = new ArrayList<Long>();
            temp.add(val);
            (stage == 1 ? charMap1 : (stage == 2?
charMap2:charMap3)).put(e.getKeyCode(),temp);
        }
    }
    /**/
    if(e.getKeyCode() == 10) {
        if(stage == 0) {
            name = typingArea.getText();
            displayMessage2("-> "+name+": "+name);
            //displayArea.setFont(boldFont);
            displayMessage("* Leila: Good to meet you, "+name+"!"+
                "\n* Leila: Let's get started, shall we?" +
                "\n* Leila: I need you to type this sentence for me:" +
                "\n "+typeText);
            stage++;
        }
        else if(stage == 1) {
            //stage = 2;
            if(Math.abs(typingArea.getText().length() - typeText.length()) < 3) {
                displayMessage2("-> "+name+": "+typingArea.getText());
                displayMessage("* Leila: Great! That was perfect." +
                    "\n* Leila: "+getr(quest_starts)+" "+getr(questions));
                stage++;
            } else
                displayMessage("* Leila: Uhm... There were too many errors, could you try
again?\n "+typeText);
        } else {
            if(t2) {
                network.train(charMap1,charMap3);
                t2 = false;
            }
            double n1v = network.eval(charMap1), n2v = network.eval(charMap3);

```

```

System.out.println("1: "+n1v+" 2: "+n2v);

displayMessage2("-> "+name+": "+typingArea.getText());
if(n1v > n2v) {
    warning++;
    displayMessage2("<< WARNING LEVEL "+(10*warning)+" >>");
}
if(warning > 9) {
    System.out.println("\nINTRUDER DETECTED. SYSTEM IS LOCKED\n");
    System.exit(0);
}

displayMessage("* Leila: "+getr(answers)+
    "\n* Leila: "+getr(quest_starts)+" "+getr(questions));
charMap3.clear();
}
/*
if(stage == 2) {
    //stage = 3;
    displayMessage("\n\nMystery typer, please type: "+typeText);
} else if(stage == 3) {
    displayMessage("done!");
    train1();
    for(Integer p : charMap1.keySet()) {
        displayArea.append(p.intValue()+" ");
        for(int i = 0; i < charMap1.get(p).size() ; i++)
            displayArea.append( charMap1.get(p).get( i ) +" ");
        displayArea.append(newline);
    }
    displayArea.setCaretPosition(displayArea.getDocument().getLength());
}
//stage++;
*/
typingArea.setText("");
}
//}
/*
//if(e.getKeyChar() == 'a') {
displayArea.append(e.getKeyChar()+" ");
for(int i = 0; i < charMap.get(e.getKeyCode()).size() ; i++)
    displayArea.append( charMap.get(e.getKeyCode()).get( i ) +" ");
displayArea.append(newline);
displayArea.setCaretPosition(displayArea.getDocument().getLength());
//}
*/
}

/** Handle the button click. */
public void actionPerformed(ActionEvent e) {

```



```

//Clear the text components.
displayArea.setText("");
typingArea.setText("");

//Return the focus to the typing area.
typingArea.requestFocusInWindow();
}

private void displayMessage2(String s){
    displayArea.append(s + "\n");
}
private void displayMessage(String s){
    //displayArea.append(s + "\n");
    /* */
    Thread thread1 = new Thread(new MessageThread(displayArea, s), "thread1");
    thread1.start();
    //displayArea.append(s + "\n");*/
}
}

```

### ***Neural Network Classes:***

**NN.java:** *This is an interface which represents a neural network. This is useful because then all of my neural network classes can implement this interface, allowing me to “black-box” the network classes and easily interchange them in my programs or switch them between programs.*

// abstract class representing a Neural Network

```

import java.util.*;
public interface NN {
    void train(Map<Integer, ArrayList<Long>> charMap1, Map<Integer, ArrayList<Long>> charMap2);
    double eval(Map<Integer, ArrayList<Long>> charMap3);
}

```

**NNSingleLayer.java:** *This is a class which embodies the functionality of a single-layered neural network, as the name suggests. It is the simplest kind of neural network, and it is specifically set up for this scenario – the typing data. This class represents a single, stand-alone neural network, which has its own weight vector values, and can be trained by submission of typing data, and then other typing data can be submitted and evaluated.*

// a simple single-layer neural network

```

import java.util.*;

public class NNSingleLayer implements NN {
    //the neural network
    Map<Integer, Double> weights = new HashMap<Integer, Double>();
    boolean setWeights = false;

    public void NNSingleLayer() {

    }
}

```

```

public void train(Map<Integer, ArrayList<Long>> charMap1, Map<Integer, ArrayList<Long>>
charMap2) {
    train1(charMap1,charMap2);
}

public void train1(Map<Integer, ArrayList<Long>> charMap1, Map<Integer, ArrayList<Long>>
charMap2) {
    System.out.println("Starting the training process...");
    //the vectors to feed into the neural network
    Map<Integer, Long> curMap;
    Map<Integer, Long> curMap1 = new HashMap<Integer,Long>();
    Map<Integer, Long> curMap2 = new HashMap<Integer,Long>();
    //making sure that both vectors have the same keys
    ArrayList<Integer> toRemove = new ArrayList<Integer>();
    for(Integer i:charMap1.keySet())
        if(!charMap2.containsKey(i))
            toRemove.add(i);
    for(Integer i:toRemove)
        charMap1.remove(i);
    toRemove.clear();
    for(Integer i:charMap2.keySet())
        if(!charMap1.containsKey(i))
            toRemove.add(i);
    for(Integer i:toRemove)
        charMap2.remove(i);
    //intializing the weights
    if(!setWeights) {
        for(Integer i:charMap2.keySet())
            weights.put(i,Math.random());
        //Should be unnecessary...
        for(Integer i:charMap1.keySet())
            weights.put(i,Math.random());
        setWeights = true;
    }

    //populate the vectors with the processed data
    long avg = 0;
    ArrayList<Long> temp = new ArrayList<Long>();
    for(Integer i:charMap1.keySet()) {
        avg = 0;
        temp = charMap1.get(i);
        for(Long l:temp)
            avg += l.longValue();
        avg /= temp.size();
        curMap1.put(i,avg);
    }
    for(Integer i:charMap2.keySet()) {
        avg = 0;
        temp = charMap2.get(i);

```

```

        for(Long l:temp)
            avg += l.longValue();
        avg /= temp.size();
        curMap2.put(i,avg);
    }

    //time to train the network
    double[] ans = {1.0,0.0};
    int curPos = 0;
    int limit = 30000;
    double dd = .1;
    System.out.println("  Network set-up is done, starting "+limit+" cycles...");
    printWeights();
    for(int j=0; j < limit; j++) {
        for(Integer i : weights.keySet()) {
            //System.out.println("aaaa");
            for(curPos = 0; curPos < 2; curPos++) {

                curMap = (curPos==0 ? curMap1 : curMap2);
                double firstError = Math.abs(ans[curPos] - evalC(curMap));

                //change the cur value
                dd = Math.random();
                double oW = weights.get(i).doubleValue(); //old weight
                double nW = weights.get(i).doubleValue() + dd; //new weight
                weights.put(i,nW);
                //evaluate
                double secondError = Math.abs(ans[curPos] - evalC(curMap));
                //change value back
                weights.put(i,oW);
                //evaluate slope of error, proceed in correct direction.
                double slope = (secondError - firstError) / dd;
                //System.out.println("First: " + firstError + " Second: "+ (secondError)
+ " Slope: "+slope + " Eval: " + eval1(weights,curMap));
                weights.put(i,oW - 0.5 * slope);
            }
        }
    }

    System.out.println("  Cycles completed, this round of training is complete.");
    printWeights();
}
public void printWeights() {
    System.out.print("  Weights--");
    for(Integer i:weights.keySet())
        System.out.print(" | "+i+": "+weights.get(i));
    System.out.println();
}
public double eval(Map<Integer, ArrayList<Long>> charMap1) {

```

```

Map<Integer, Long> curMap = new HashMap<Integer,Long>();

//populate the vectors with the processed data
long avg = 0;
ArrayList<Long> temp = new ArrayList<Long>();
for(Integer i:charMap1.keySet()) {
    avg = 0;
    temp = charMap1.get(i);
    for(Long l:temp)
        avg += l.longValue();
    avg /= temp.size();
    curMap.put(i,avg);
}

//calculate the sum
double ans = 0.0;
for(Integer i: weights.keySet()) {
    if(curMap.containsKey(i)) {
        ans += weights.get(i).doubleValue() * curMap.get(i).longValue();
    }
}
ans /= curMap.size();
//activation function
ans = 1.0 / (1+Math.pow(2.718, -.001 * ans));
return ans;
}

public double evalC(Map<Integer, Long> curMap) {
    //calculate the sum
    double ans = 0.0;
    for(Integer i: weights.keySet()) {
        if(curMap.containsKey(i)) {
            ans += weights.get(i).doubleValue() * curMap.get(i).longValue();
        }
    }
}

// System.out.println(ans);
//activation function
ans /= curMap.size();
ans = 1.0 / (1+Math.pow(2.718, -.001 * ans));
return ans;
}
}

```

**NNMultiLayer.java:** *This is a class which embodies the functionality of a multi-layered neural network, as the name suggests. The number of layers is dynamic and can be determined by the program which creates an instance of this class. It is a more complex kind of neural network, and it is specifically set up for this scenario – the typing data. This class represents a single, stand-alone neural network, which has its own weight vector values, and can be trained by submission of typing data, and then other typing data can be submitted and evaluated.*

// a simple single-layer neural network

```

import java.util.*;

public class NNMultiLayer implements NN {
    //the neural network
    ArrayList<Map<Integer, Double>> weights = new ArrayList<Map<Integer, Double>>();
    ArrayList<ArrayList<Double>> comp = new ArrayList<ArrayList<Double>>();
    boolean setWeights = false;

    public NNMultiLayer(int numLayers) {
        for(int i=0; i<numLayers; i++)
            weights.add(new HashMap<Integer, Double>());
    }

    public void train(Map<Integer, ArrayList<Long>> charMap1, Map<Integer, ArrayList<Long>>
charMap2) {
        train1(charMap1,charMap2);
    }

    public void train1(Map<Integer, ArrayList<Long>> charMap1, Map<Integer, ArrayList<Long>>
charMap2) {
        System.out.println("Starting the training process...");
        //the vectors to feed into the neural network
        Map<Integer, Long> curMap;
        Map<Integer, Long> curMap1 = new HashMap<Integer,Long>();
        Map<Integer, Long> curMap2 = new HashMap<Integer,Long>();
        //making sure that both vectors have the same keys
        ArrayList<Integer> toRemove = new ArrayList<Integer>();
        for(Integer i:charMap1.keySet())
            if(!charMap2.containsKey(i))
                toRemove.add(i);
        for(Integer i:toRemove)
            charMap1.remove(i);
        toRemove.clear();
        for(Integer i:charMap2.keySet())
            if(!charMap1.containsKey(i))
                toRemove.add(i);
        for(Integer i:toRemove)
            charMap2.remove(i);
        //intializing the weights
        if(!setWeights) {
            for(Integer i:charMap2.keySet())
                weights.put(i,Math.random());
            //Should be unnecessary...
            for(Integer i:charMap1.keySet())
                weights.put(i,Math.random());
            setWeights = true;
        }

        //populate the vectors with the processed data
    }
}

```

```

long avg = 0;
ArrayList<Long> temp = new ArrayList<Long>();
for(Integer i:charMap1.keySet()) {
    avg = 0;
    temp = charMap1.get(i);
    for(Long l:temp)
        avg += l.longValue();
    avg /= temp.size();
    curMap1.put(i,avg);
}
for(Integer i:charMap2.keySet()) {
    avg = 0;
    temp = charMap2.get(i);
    for(Long l:temp)
        avg += l.longValue();
    avg /= temp.size();
    curMap2.put(i,avg);
}

//time to train the network
double[] ans = {1.0,0.0};
int curPos = 0;
int limit = 30000;
double dd = .1;
System.out.println("  Network set-up is done, starting "+limit+" cycles...");
printWeights();
for(int j=0; j < limit; j++) {
    for(Integer i : weights.keySet()) {
        //System.out.println("aaaa");
        for(curPos = 0; curPos < 2; curPos++) {

            curMap = (curPos==0 ? curMap1 : curMap2);
            double firstError = Math.abs(ans[curPos] - evalC(curMap));

            //change the cur value
            dd = Math.random();
            double oW = weights.get(i).doubleValue(); //old weight
            double nW = weights.get(i).doubleValue() + dd; //new weight
            weights.put(i,nW);
            //evaluate
            double secondError = Math.abs(ans[curPos] - evalC(curMap));
            //change value back
            weights.put(i,oW);
            //evaluate slope of error, proceed in correct direction.
            double slope = (secondError - firstError) / dd;
            //System.out.println("First: " + firstError + " Second: "+ (secondError)
+ " Slope: "+slope + " Eval: " + eval1(weights,curMap));
            weights.put(i,oW - 0.5 * slope);
        }
    }
}

```

```

        }
    }

    System.out.println("    Cycles completed, this round of training is complete.");
    printWeights();
}
// public void printWeights() {
//     System.out.print("    Weights--");
//     for(Integer i:weights.keySet())
//         System.out.print(" | "+i+": "+weights.get(i));
//     System.out.println();
// }
public double eval(Map<Integer, ArrayList<Long>> charMap1) {
    Map<Integer, Long> curMap = new HashMap<Integer,Long>();

    //populate the vectors with the processed data
    long avg = 0;
    ArrayList<Long> temp = new ArrayList<Long>();
    for(Integer i:charMap1.keySet()) {
        avg = 0;
        temp = charMap1.get(i);
        for(Long l:temp)
            avg += l.longValue();
        avg /= temp.size();
        curMap.put(i,avg);
    }

    //calculate the sum
    double ans = 0.0;
    for(Integer i: weights.keySet()) {
        if(curMap.containsKey(i)) {
            ans += weights.get(i).doubleValue() * curMap.get(i).longValue();
        }
    }
    ans /= curMap.size();
    //activation function
    ans = 1.0 / (1+Math.pow(2.718, -.001 * ans));
    return ans;
}
public double evalC(Map<Integer, Long> curMap) {
    //calculate the sum
    double ans = 0.0;
    comp.clear();
    for(int i=0; i<weights.size(); i++)
        comp.add(new ArrayList<Double>());
    int level = 0;
    for(Map<Integer, Double> m: weights) {
        int pos = 0;
        for(Integer i: m.keySet()) {

```

```
        }
        if(curMap.containsKey(i)) {
            ans += weights.get(i).doubleValue() * curMap.get(i).longValue();
        }
    }
//    System.out.println(ans);
//activation function
ans /= curMap.size();
ans = 1.0 / (1+Math.pow(2.718, -.001 * ans));
return ans;
}
}
```



## Q1Driver.java

*This code is the proof of concept application that I programmed in the first quarter. It displays a text box, prompts two users to both type a sentence, measures their typing characteristics, trains a neural network with their keystroke data, prompts one of the users (identity unknown to the computer) to type a third sentence, measures that user's keystroke data, runs that data through the neural network, and outputs whether the third typist was the first or second typist.*

```
// Dynamic Authentication by Typing Patterns
// 1st Quarter Program Code
// Luke Knepper, 2009
// luke@lukeknepper.com
// *All rights reserved.*
```

```
import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.Dimension;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;
```

```
public class Driver extends JFrame
    implements KeyListener,
    ActionListener
```

```
{
    JTextArea displayArea;
    JTextField typingArea;
    static final String newline = System.getProperty("line.separator");
    //Format is KeyCode, data
    static Map<Integer, ArrayList<Long>> charMap1 = new HashMap<Integer, ArrayList<Long>>();
    static Map<Integer, ArrayList<Long>> charMap2 = new HashMap<Integer, ArrayList<Long>>();
    static Map<Integer, ArrayList<Long>> charMap3 = new HashMap<Integer, ArrayList<Long>>();
    Map<Integer, Long> curKeyTimes = new HashMap<Integer, Long>();
    //String typeText = "the quick brown fox jumps over the lazy brown dog on a saturday afternoon in
brooklyn near the bay bridge without coughing or stopping at the crosswalk and then takes a bite of pizza
without paying any money for it and then goes home and sleeps <ENTER>";
    String typeText = "the quick brown fox jumps over the lazy dog\n <ENTER>\n";
    int curTry = 0;
```

```
public static void train1() {
```

```
    Map<Integer, Long> curMap1 = new HashMap<Integer,Long>();
    Map<Integer, Long> curMap2 = new HashMap<Integer,Long>();
    Map<Integer, Long> curMap3 = new HashMap<Integer,Long>();
```

```
    for(Integer i:charMap1.keySet()) {
        long avg = 0;
        for(Long l:charMap1.get(i))
            avg += l.longValue();
        avg /= charMap1.get(i).size();
```

```

//System.out.println(avg);
curMap1.put(i,avg);
//System.out.println("aaa "+i.intValue());
}

for(Integer i:charMap2.keySet()) {
    long avg = 0;
    for(Long l:charMap2.get(i))
        avg += l.longValue();
    avg /= charMap2.get(i).size();
    //System.out.println(avg);
    curMap2.put(i,avg);
}

for(Integer i:charMap3.keySet()) {
    long avg = 0;
    for(Long l:charMap3.get(i))
        avg += l.longValue();
    avg /= charMap3.get(i).size();
    //System.out.println(avg);
    curMap3.put(i,avg);
}

Map<Integer, Long> curMap = curMap2;

//double[] weights = new double[256];
Map<Integer, Double> weights = new HashMap<Integer, Double>();
for(Integer i:charMap2.keySet())
    weights.put(i,Math.random());
for(Integer i:charMap1.keySet())
    weights.put(i,Math.random());
for(Integer i:charMap3.keySet())
    weights.put(i,Math.random());

double[] ans = {1.0,0.0};
int curPos = 0;
int limit = 10;
double dd = .1;
for(int j=0; j < limit; j++) {
    for(Integer i : weights.keySet()) {
        //System.out.println("aaaa");
        for(curPos = 0; curPos < 2; curPos++) {
            curMap = (curPos==0 ? curMap1 : curMap2);
            double firstError = Math.abs(ans[curPos] - eval1(weights,curMap));
            //dd = Math.random();

            //change the cur value
            dd = Math.random();
            double oW = weights.get(i).doubleValue(); //old weight

```

```

        double nW = weights.get(i).doubleValue() + dd; //new weight
        weights.put(i,nW);
        //evaluate
        double secondError = Math.abs(ans[curPos] - eval1(weights,curMap));
        //change value back
        weights.put(i,oW);
        //evaluate slope of error, proceed in correct direction.
        double slope = (secondError - firstError) / dd;
        System.out.println("First: " + firstError + " Second: " + (secondError) +
" Slope: "+slope + " Eval: " + eval1(weights,curMap));
        weights.put(i,oW + 0.5 * slope);
    }
}
}

```

```

System.out.println("Finals:");
System.out.println("1: "+eval1(weights,curMap1));
System.out.println("2: "+eval1(weights,curMap2));
double firstOff = Math.abs(eval1(weights,curMap1) - eval1(weights,curMap3));
double secondOff = Math.abs(eval1(weights,curMap2) - eval1(weights,curMap3));
System.out.println("1e: "+firstOff);
System.out.println("2e: "+secondOff);

```

```

System.out.println("\n\nYou are typer "+(firstOff < secondOff? "one" : "two")+"!");
}

```

```

public static double eval1(Map<Integer,Double> weights, Map<Integer, Long> curMap) {
    double total= 0.0;
    for(Integer i:curMap.keySet()) {
        total += weights.get(i).doubleValue() * curMap.get(i).longValue();
    }
    total /= weights.size();
    //System.out.println(total);
    total = 1.0 / (1+Math.pow(2.718,(-.01 * total)));
    //System.out.println(total+"\n");
    return total;
}

```

```

public static void main(String[] args) {
    /* Use an appropriate Look and Feel */
    try {
        UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
    } catch (UnsupportedLookAndFeelException ex) {
        ex.printStackTrace();
    } catch (IllegalAccessException ex) {
        ex.printStackTrace();
    } catch (InstantiationException ex) {
        ex.printStackTrace();
    }
}

```

```

    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    }
    /* Turn off metal's use of bold fonts */
    UIManager.put("swing.boldMetal", Boolean.FALSE);

    //Schedule a job for event dispatch thread:
    //creating and showing this application's GUI.
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}

/**
 * Create the GUI and show it. For thread safety,
 * this method should be invoked from the
 * event-dispatching thread.
 */
private static void createAndShowGUI() {
    //Create and set up the window.
    Driver frame = new Driver("Super Password Data Collection");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Set up the content pane.
    frame.addComponentsToPane();

    //Display the window.
    frame.pack();
    frame.setVisible(true);
}

private void addComponentsToPane() {

    JButton button = new JButton("Clear");
    button.addActionListener(this);

    typingArea = new JTextField(20);
    typingArea.addKeyListener(this);

    //Uncomment this if you wish to turn off focus
    //traversal. The focus subsystem consumes
    //focus traversal keys, such as Tab and Shift Tab.
    //If you uncomment the following line of code, this
    //disables focus traversal and the Tab events will
    //become available to the key event listener.
    //typingArea.setFocusTraversalKeysEnabled(false);

```

```

displayArea = new JTextArea();
displayArea.setEditable(false);
JScrollPane scrollPane = new JScrollPane(displayArea);
scrollPane.setPreferredSize(new Dimension(375, 125));

getContentPane().add(typingArea, BorderLayout.PAGE_START);
getContentPane().add(scrollPane, BorderLayout.CENTER);
getContentPane().add(button, BorderLayout.PAGE_END);
}

public Driver(String name) {
    super(name);
}

/** Handle the key typed event from the text field. */
public void keyTyped(KeyEvent e) {
    //displayInfo(e, "KEY TYPED: ");
}

/** Handle the key pressed event from the text field. */
public void keyPressed(KeyEvent e) {
    //displayInfo(e, "KEY PRESSED: ");
    //add key info and time to the curKeyTimes map
    curKeyTimes.put(e.getKeyCode(), System.currentTimeMillis());

    //if(curKeyTimes.containsKey(e.getKeyCode())) {
    //    curKeyTimes.
}

/** Handle the key released event from the text field. */
public void keyReleased(KeyEvent e) {
    //displayInfo(e, "KEY RELEASED: ");
    //take key info from curKeyTimes map and put into charMap
    long val = System.currentTimeMillis() - curKeyTimes.get(e.getKeyCode());
    if(curTry <=3 ) {
        if((curTry == 1 ? charMap1 : (curTry == 2?
charMap2:charMap3)).containsKey(e.getKeyCode()))
            (curTry == 1 ? charMap1 : (curTry == 2?
charMap2:charMap3)).get(e.getKeyCode()).add(val);
        else {
            ArrayList<Long> temp = new ArrayList<Long>();
            temp.add(val);
            (curTry == 1 ? charMap1 : (curTry == 2?
charMap2:charMap3)).put(e.getKeyCode(),temp);
        }
    }
}

```

```

if(e.getKeyCode() == 10) {
    if(curTry == 0) {

        displayMessage("\n\nTyper #1, please type:\n"+typeText);
    }
    if(curTry == 1) {
        //curTry = 2;
        displayMessage("\n\nTyper #1 complete, typer #2 please type:\n"+typeText);
    } else if(curTry == 2) {
        //curTry = 3;
        displayMessage("\n\nMystery typer, please type: "+typeText);
    } else if(curTry == 3) {
        displayMessage("done!");
        train1();
        for(Integer p : charMap1.keySet()) {
            displayArea.append(p.intValue()+" ");
        }
        for(int i = 0; i < charMap1.get(p).size() ; i++)
            displayArea.append( charMap1.get(p).get( i ) +", ");
        displayArea.append(newline);
    }
}
displayArea.setCaretPosition(displayArea.getDocument().getLength());
}
curTry++;
}
}
/*
//if(e.getKeyChar() == 'a') {
displayArea.append(e.getKeyChar()+" ");
for(int i = 0; i < charMap.get(e.getKeyCode()).size() ; i++)
    displayArea.append( charMap.get(e.getKeyCode()).get( i ) +", ");
displayArea.append(newline);
displayArea.setCaretPosition(displayArea.getDocument().getLength());
//}
*/
}

/** Handle the button click. */
public void actionPerformed(ActionEvent e) {
    //Clear the text components.
    displayArea.setText("");
    typingArea.setText("");

    //Return the focus to the typing area.
    typingArea.requestFocusInWindow();
}

private void displayMessage(String s){

    displayArea.append("\n" + s + "\n");
}

```

```

    }

/*
 * We have to jump through some hoops to avoid
 * trying to print non-printing characters
 * such as Shift. (Not only do they not print,
 * but if you put them in a String, the characters
 * afterward won't show up in the text area.)
 */
private void displayInfo(KeyEvent e, String keyStatus){

    //You should only rely on the key char if the event
    //is a key typed event.
    int id = e.getID();
    String keyString;
    if (id == KeyEvent.KEY_TYPED) {
        char c = e.getKeyChar();
        keyString = "key character = " + c + " (v=" + Character.getNumericValue(c)+") ";
    } else {
        int keyCode = e.getKeyCode();
        keyString = "key code = " + keyCode
            + " ("
            + KeyEvent.getKeyText(keyCode)
            + ") " + ((char)keyCode);
    }

    int modifiersEx = e.getModifiersEx();
    String modString = "extended modifiers = " + modifiersEx;
    String tmpString = KeyEvent.getModifiersExText(modifiersEx);
    if (tmpString.length() > 0) {
        modString += " (" + tmpString + ")";
    } else {
        modString += " (no extended modifiers)";
    }

    String actionString = "action key? ";
    if (e.isActionKey()) {
        actionString += "YES";
    } else {
        actionString += "NO";
    }

    String locationString = "key location: ";
    int location = e.getKeyLocation();
    if (location == KeyEvent.KEY_LOCATION_STANDARD) {
        locationString += "standard";
    } else if (location == KeyEvent.KEY_LOCATION_LEFT) {
        locationString += "left";
    } else if (location == KeyEvent.KEY_LOCATION_RIGHT) {

```

```
    locationString += "right";
} else if (location == KeyEvent.KEY_LOCATION_NUMPAD) {
    locationString += "numpad";
} else { // (location == KeyEvent.KEY_LOCATION_UNKNOWN)
    locationString += "unknown";
}

displayArea.append(keyStatus + newline
    + " " + keyString + newline
    + " " + modString + newline
    + " " + actionString + newline
    + " " + locationString + newline);
displayArea.setCaretPosition(displayArea.getDocument().getLength());
}
}
```



## **DataGather fla**

*This is the code from the data gathering application that I coded in Flash. It's purpose is to collect keystroke data from many people via an online applet. It displays a text box, asks the user to type in two sentences, measures their keystroke data, and passes that data on to DataSave.php when they are done.*

var longText:String = " To Sherlock Holmes she is always the woman. I have seldom heard him mention her under any other name. In his eyes she eclipses and predominates the whole of her sex. It was not that he felt any emotion akin to love for Irene Adler. All emotions, and that one particularly, were abhorrent to his cold, precise, but admirably balanced mind. He was, I take it, the most perfect reasoning and observing machine that the world has seen; but, as a lover, he would have placed himself in a false position. He never spoke of the softer passions, save with a gibe and a sneer. They were admirable things for the observer excellent for drawing the veil from men's motives and actions. But for the trained reasoner to admit such intrusions into his own delicate and finely adjusted temperament was to introduce a distracting factor which might throw a doubt upon all his mental results. Grit in a sensitive instrument, or a crack in one of his own high-power lenses, would not be more disturbing than a strong emotion in a nature such as his. And yet there was but one woman to him, and that woman was the late Irene Adler, of dubious and questionable memory. I had seen little of Holmes since the singular chain of events which I have already narrated in a bold fashion under the heading of The Sign of Four. My marriage had, as he foretold, drifted us away from each other. My own complete happiness, and the home-centred interests which rise up around the man who first finds himself master of his own establishment, were sufficient to absorb all my attention; while Holmes, who loathed every form of society with his whole Bohemian soul, remained in our lodgings in Baker Street, buried among his old books, and alternating from week to week between cocaine and ambition, the drowsiness of the drug, and the fierce energy of his own keen nature. He was still, as ever, deeply attracted by the study of crime, and occupied his immense faculties and extraordinary powers of observation in following out those clues, and clearing up those mysteries, which had been abandoned as hopeless by the official police. From time to time I heard some vague account of his doings: of his summons to Odessa in the case of the Trepoff murder, of his clearing up of the singular tragedy of the Atkinson brothers at Trincomalee, and finally of the mission which he had accomplished so delicately and successfully for the reigning family of Holland. Beyond these signs of his activity, however, which I merely shared with all the readers of the daily press, I knew little of my former friend and companion. One night it was on the 20th of March, 1888, I was returning from a journey to a patient (for I had now returned to civil practice), when my way led me through Baker Street. As I passed the well-remembered door, which must always be associated in my mind with my wooing, and with the dark incidents of the Study in Scarlet, I was seized with a keen desire to see Holmes again, and to know how he was employing his extraordinary powers. His rooms were brilliantly lit, and, even as I looked up, I saw his tall spare figure pass twice in a dark silhouette against the blind. He was pacing the room swiftly, eagerly, with his head sunk upon his chest, and his hands clasped behind him. To me, who knew his every mood and habit, his attitude and manner told their own story. He was at work again. He had risen out of his drug-created dreams, and was hot upon the scent of some new problem. I rang the bell, and was shown up to the chamber which had formerly been in part my own. His manner was not effusive. It seldom was; but he was glad, I think, to see me. With hardly a word spoken, but with a kindly eye, he waved me to an armchair, threw across his case of cigars, and indicated a spirit case and a gasogene in the corner. Then he stood before the fire, and looked me over in his singular introspective fashion. A man entered who could hardly have been less than six feet six inches in height, with the chest and limbs of a Hercules. His dress was rich with a richness which would, in England, be looked upon as akin to bad taste. Heavy bands of astrakhan were slashed across the sleeves and fronts of his double-breasted coat, while the deep blue cloak which was thrown over his shoulders was lined with flame-coloured silk, and secured at the neck with a brooch which consisted of a single flaming beryl. Boots which extended half-way up his calves, and which were trimmed at the tops with a rich brown fur, completed the impression of barbaric opulence which was suggested by his

whole appearance. He carried a broad-brimmed hat in his hand, while he wore across the upper part of his face, extending down past the cheek-bones, a black vizard mask, which he had apparently adjusted that very moment, for his hand was still raised to it as he entered. From the lower part of the face he appeared to be a man of strong character, with a thick, hanging lip, and a long straight chin, suggestive of resolution pushed to the length of obstinacy. At three o'clock precisely I was at Baker Street, but Holmes had not yet returned. The landlady informed me that he had left the house shortly after eight o'clock in the morning. I sat down beside the fire, however, with the intention of awaiting him, however long he might be. I was already deeply interested in his inquiry, for, though it was surrounded by none of the grim and strange features which were associated with the two crimes which I have elsewhere recorded, still, the nature of the case and the exalted station of his client gave it a character of its own. Indeed, apart from the nature of the investigation which my friend had on hand, there was something in his masterly grasp of a situation, and his keen, incisive reasoning, which made it a pleasure to me to study his system of work, and to follow the quick, subtle methods by which he disentangled the most inextricable mysteries. So accustomed was I to his invariable success that the very possibility of his failing had ceased to enter into my head. I was half-dragged up to the altar, and before I knew where I was, I found myself mumbling responses which were whispered in my ear, and vouching for things of which I knew nothing, and generally assisting in the secure tying up of Irene Adler, spinster, to Godfrey Norton, bachelor. It was all done in an instant, and there was the gentleman thanking me on the one side and the lady on the other, while the clergyman beamed on me in front. It was the most preposterous position in which I ever found myself in my life, and it was the thought of it that started me laughing just now. It seems that there had been some informality about their licence, that the clergyman absolutely refused to marry them without a witness of some sort, and that my lucky appearance saved the bridegroom from having to sally out into the streets in search of a best man. The bride gave me a sovereign, and I mean to wear it on my watch-chain in memory of the occasion. It was a quarter past six when we left Baker Street, and it still wanted ten minutes to the hour when we found ourselves in Serpentine Avenue. It was already dusk, and the lamps were just being lighted as we paced up and down in front of Briony Lodge, waiting for the coming of its occupant. The house was just such as I had pictured it from Sherlock Holmes's succinct description, but the locality appeared to be less private than I expected. On the contrary, for a small street in a quiet neighbourhood, it was remarkably animated. There was a group of shabbily-dressed men smoking and laughing in a corner, a scissors-grinder with his wheel, two guardsmen who were flirting with a nurse-girl, and several well-dressed young men who were lounging up and down with cigars in their mouths. As he spoke, the gleam of the sidelights of a carriage came round the curve of the avenue. It was a smart little landau which rattled up to the door of Briony Lodge. As it pulled up, one of the loafing men at the corner dashed forward to open the door in the hope of earning a copper, but was elbowed away by another loafer who had rushed up with the same intention. A fierce quarrel broke out, which was increased by the two guardsmen, who took sides with one of the loungers, and by the scissors-grinder, who was equally hot upon the other side. A blow was struck, and in an instant the lady, who had stepped from her carriage, was the centre of a little knot of flushed and struggling men who struck savagely at each other with their fists and sticks. Holmes dashed into the crowd to protect the lady; but just as he reached her, he gave a cry and dropped to the ground, with the blood running freely down his face. At his fall the guardsmen took to their heels in one direction and the loungers in the other, while a number of better-dressed people who had watched the scuffle without taking part in it, crowded in to help the lady and to attend to the injured man. Irene Adler, as I will still call her, had hurried up the steps; but she stood at the top with her superb figure outlined against the lights of the hall, looking back into the street. Slowly and solemnly he was borne into Briony Lodge, and laid out in the principal room, while I still observed the proceedings from my post by the window. The lamps had been lit, but the blinds had not been drawn, so that I could see Holmes as he lay upon the couch. I do not know whether he was seized with compunction at that moment for the part he was playing, but I know that I never felt more heartily ashamed of myself in my life than when I saw the beautiful creature against whom I was conspiring, or the grace and kindness with which she waited upon the injured man. And yet it would be the blackest treachery to Holmes to draw back now from the part which he had

entrusted to me. I hardened my heart and took the smoke-rocket from under my ulster. After all, I thought, we are not injuring her. We are but preventing her from injuring another. Holmes had sat up upon the couch, and I saw him motion like a man who is in want of air. A maid rushed across and threw open the window. At the same instant I saw him raise his hand, and at the signal I tossed my rocket into the room with a cry of fire. The word was no sooner out of my mouth than the whole crowd of spectators, well dressed and ill gentlemen, ostlers, and servant maids joined in a general shriek of fire. Thick clouds of smoke curled into the room, and out at the open window. I caught a glimpse of rushing figures, and a moment later the voice of Holmes from within, assuring them that it was a false alarm. Slipping through the shouting crowd I made my way to the corner of the street, and in ten minutes was rejoiced to find my friend's arm in mine, and to get away from the scene of the uproar. He walked swiftly and in silence for some few minutes, until we had turned down one of the quiet streets which lead towards the Edgware Road. It was all-important. When a woman thinks that her house is on fire, her instinct is at once to rush to the thing which she values most. It is a perfectly overpowering impulse, and I have more than once taken advantage of it. In the case of the Darlington Substitution Scandal it was of use to me, and also in the Arnsworth Castle business. A married woman grabs at her baby. An unmarried one reaches for her jewel box. Now it was clear to me that our lady of to-day had nothing in the house more precious to her than what we are in quest of. She would rush to secure it. The alarm of fire was admirably done. The smoke and shouting was enough to shake nerves of steel. She responded beautifully. The photograph is in a recess behind a sliding panel just above the right bell-pull. She was there in an instant, and I caught a glimpse of it as she half drew it out. When I cried out that it was a false alarm, she replaced it, glanced at the rocket, rushed from the room, and I have not seen her since. I rose, and, making my excuses, escaped from the house. I hesitated whether to attempt to secure the photograph at once; but the coachman had come in, and as he was watching me narrowly, it seemed safer to wait. A little over-precipitance may ruin all. You really did it very well. You took me in completely. Until after the alarm of fire, I had not a suspicion. But then, when I found how I had betrayed myself, I began to think. I had been warned against you months ago. I had been told that if the King employed an agent, it would certainly be you. And your address had been given me. Yet, with all this, you made me reveal what you wanted to know. Even after I became suspicious, I found it hard to think evil of such a dear, kind old clergyman. But, you know, I have been trained as an actress myself. Male costume is nothing new to me. I often take advantage of the freedom which it gives. I sent John, the coachman, to watch you, ran upstairs, got into my walking clothes, as I call them, and came down just as you departed. He had risen from his chair, and was standing between the parted blinds, gazing down into the dull, neutral-tinted London street. Looking over his shoulder I saw that on the pavement opposite there stood a large woman with a heavy fur boa round her neck, and a large curling red feather in a broad-brimmed hat which was tilted in a coquettish Duchess-of-Devonshire fashion over her ear. From under this great panoply she peeped up in a nervous, hesitating fashion at our windows, while her body oscillated backwards and forwards, and her fingers fidgeted with her glove buttons. Suddenly, with a plunge, as of the swimmer who leaves the bank, she hurried across the road, and we heard the sharp clang of the bell. All children, except one, grow up. They soon know that they will grow up, and the way Wendy knew was this. One day when she was two years old she was playing in a garden, and she plucked another flower and ran with it to her mother. I suppose she must have looked rather delightful, for Mrs Darling put her hand to her heart and cried, Oh, why can't you remain like this for ever! This was all that passed between them on the subject, but henceforth Wendy knew that she must grow up. You always know after you are two. Two is the beginning of the end. Of course they lived at 14, and until Wendy came her mother was the chief one. She was a lovely lady, with a romantic mind and such a sweet mocking mouth. Her romantic mind was like the tiny boxes, one within the other, that come from the puzzling East, however many you discover there is always one more; and her sweet mocking mouth had one kiss on it that Wendy could never get, though there it was, perfectly conspicuous in the right-hand corner. The way Mr Darling won her was this: the many gentlemen who had been boys when she was a girl discovered simultaneously that they loved her, and they all ran to her house to propose to her except Mr Darling, who took a cab and nipped in first, and so he got her. He got all of her, except the innermost box and the kiss. He never knew

about the box, and in time he gave up trying for the kiss. Wendy thought Napoleon could have got it, but I can picture him trying, and then going off in a passion, slamming the door. Mr Darling used to boast to Wendy that her mother not only loved him but respected him. He was one of those deep ones who know about stocks and shares. Of course no one really knows, but he quite seemed to know, and he often said stocks were up and shares were down in a way that would have made any woman respect him. Mrs Darling was married in white, and at first she kept the books perfectly, almost gleefully, as if it were a game, not so much as a brussels sprout was missing; but by and by whole cauliflowers dropped out, and instead of them there were pictures of babies without faces. She drew them when she should have been totting up. They were Mrs Darling's guesses. Mrs Darling loved to have everything just so, and Mr Darling had a passion for being exactly like his neighbours; so, of course, they had a nurse. As they were poor, owing to the amount of milk the children drank, this nurse was a prim Newfoundland dog, called Nana, who had belonged to no one in particular until the Darlings engaged her. She had always thought children important, however, and the Darlings had become acquainted with her in Kensington Gardens, where she spent most of her spare time peeping into perambulators, and was much hated by careless nursemaids, whom she followed to their homes and complained of to their mistresses. She proved to be quite a treasure of a nurse. How thorough she was at bath-time; and up at any moment of the night if one of her charges made the slightest cry. Of course her kennel was in the nursery. She had a genius for knowing when a cough is a thing to have no patience with and when it needs stocking round your throat. She believed to her last day in old-fashioned remedies like rhubarb leaf, and made sounds of contempt over all this new-fangled talk about germs, and so on. It was a lesson in propriety to see her escorting the children to school, walking sedately by their side when they were well behaved, and butting them back into line if they strayed. On John's footer days she never once forgot his sweater, and she usually carried an umbrella in her mouth in case of rain. There is a room in the basement of Miss Fulsom's school where the nurses wait. They sat on forms, while Nana lay on the floor, but that was the only difference. They affected to ignore her as of an inferior social status to themselves, and she despised their light talk. She resented visits to the nursery from Mrs Darling's friends, but if they did come she first whipped off Michael's pinafore and put him into the one with blue braiding, and smoothed out Wendy and made a dash at John's hair. Nana also troubled him in another way. He had sometimes a feeling that she did not admire him. I know she admires you tremendously, George, Mrs Darling would assure him, and then she would sign to the children to be specially nice to father. Lovely dances followed, in which the only other servant, Liza, was sometimes allowed to join. Such a midget she looked in her long skirt and maid's cap, though she had sworn, when engaged, that she would never see ten again. The gaiety of those romps! And gayest of all was Mrs Darling, who would pirouette so wildly that all you could see of her was the kiss, and then if you had dashed at her you might have got it. There never was a simpler, happier family until the coming of Peter Pan. Mrs Darling first heard of Peter when she was tidying up her children's minds. It is the nightly custom of every good mother after her children are asleep to rummage in their minds and put things straight for next morning, repacking into their proper places the many articles that have wandered during the day. If you could keep awake (but of course you can't) you would see your own mother doing this, and you would find it very interesting to watch her. It is quite like tidying up drawers. You would see her on her knees, I expect, lingering humorously over some of your contents, wondering where on earth you had picked this thing up, making discoveries sweet and not so sweet, pressing this to her cheek as if it were as nice as a kitten, and hurriedly stowing that out of sight. When you wake in the morning, the naughtinesses and evil passions with which you went to bed have been folded up small and placed at the bottom of your mind; and on the top, beautifully aired, are spread out your prettier thoughts, ready for you to put on. I don't know whether you have ever seen a map of a person's mind. Doctors sometimes draw maps of other parts of you, and your own map can become intensely interesting, but catch them trying to draw a map of a child's mind, which is not only confused, but keeps going round all the time. There are zigzag lines on it, just like your temperature on a card, and these are probably roads in the island; for the Neverland is always more or less an island, with astonishing splashes of colour here and there, and coral reefs and rakish-looking craft in the offing, and savages and lonely lairs, and gnomes who are mostly tailors, and caves through which a river

runs, and princes with six elder brothers, and a hut fast going to decay, and one very small old lady with a hooked nose. It would be an easy map if that were all; but there is also first day at school, religion, fathers, the Round Pond, needlework, murders, hangings, verbs that take the dative, chocolate-pudding day, getting into braces, say ninety-nine, threepence for pulling out your tooth yourself, and so on; and either these are part of the island or they are another map showing through, and it is all rather confusing, especially as nothing will stand still. Of course the Neverlands vary a good deal. John's, for instance, had a lagoon with flamingoes flying over it at which John was shooting, while Michael, who was very small, had a flamingo with lagoons flying over it. John lived in a boat turned upside down on the sands, Michael in a wigwam, Wendy in a house of leaves deftly sewn together. John had no friends, Michael had friends at night, Wendy had a pet wolf forsaken by its parents; but on the whole the Neverlands have a family resemblance, and if they stood still in a row you could say of them that they have each other's nose, and so forth. On these magic shores children at play are for ever beaching their coracles. We too have been there; we can still hear the sound of the surf, though we shall land no more. Centuries of time went by, and the generations of man continually increased on the earth. They scattered over wider and wider tracts of country, venturing on into regions until then strange and untrodden. There were some who lived a life of continual roving and wandering. They pitched their tents in the wild as fancy led. Others found good pastures and dwelt there, tilling the ground and gathering together flocks of sheep and herds of cattle. Yet others reared up cities, and walled them in and fortified them against their foes. And they set up kings over them, mighty in pride and soldiery and armed with weapons of war. They learned, too, the skill of many handicrafts and how to work in metals. They fashioned instruments of music, for dancing and feasting. They made wine out of the grape and were merry. And the daughters of men were fair as the morning. They walked in their beauty like barbaric queens, bedecked with fine raiment and jewels of gold and coloured gems. In these days men lived to a great age, and amassed knowledge and discovered secret arts and became practised in magic, and were wise in their own eyes. But though there was no end to the skill and invention and curiosity of their minds, the spirit of life within them languished as if in a prison-house, and was darkened. The knowledge of what is good and what evil was theirs. They were free to make choice between them. They chose evil and not good, and refused the Lord God their love and obedience. Pitiless and defiant, wherever they went, greed and violence and cruelty went with them, and no man was safe. They not only did evil, but in heart and imagination hated and fought against the good. The memory of the paradise that had been made for man had become less than the substance of a dream. And when, in despair at the defeat of their wild desires, its vision returned to them, they mocked it down and reviled the very thought of it. Angels fallen from grace entered in upon the earth in those days, and there were tyrants and giants in the land, terrible and mighty. Human life had become a mockery and a snare, because of the vileness of the spirit within. And the Lord God, looking down from the heavens upon the earth which he had created, once radiant with light and peace and innocence, and now a waste of sin and woe, repented him that he had given life to the dust. He was grieved to the heart that man, whom he had made in his own image and of his divine love, had fallen to a state so dark that even the hope and desire of goodness had perished in him. And the Lord God said: I will do away man, whom I made out of nought, from the face of the earth, and all things that have life; for it repenteth me that I created them. One alone of all men living found grace in his eyes. He was faithful and blameless. Loving goodness and hating evil, he had withdrawn himself from his fellow-men and lived apart from them; and the Lord God was with him in the silence and secrecy of his heart. The name of this man was Noah, and he had three sons, Shem, Ham and Japheth. There came a day when the Lord God warned Noah that an end was soon to come to the evil that man had brought upon the earth, and that it should be cleansed of its wickedness and corruption. And he bade Noah build an ark, or ship, directing him in the knowledge and understanding of his mind how in all things it should be made. As the Lord God bade him, so did Noah. He chose out and felled cypress and pine for timber, and with his three sons laboured day by day, until night came down and they went to rest, to fashion and build the ark, though as yet they knew neither its use nor its purpose. In length this ark was to be three hundred cubits, a hundred and fifty human paces from end to end. In breadth it was to be fifty cubits; and in height thirty of such a height, that is, that the top-most branches of an oak tree would show green

above its roof."

```
var subbed:Boolean = false;
var myString:String = longText.substring(int(Math.random()*longText.length));
myString = myString.substring(myString.indexOf('.')+2);
var t_String:String = myString.substring(200);
myString = myString.substring(0,201+t_String.indexOf('.'));
read_area.htmlText = "<b>Please type the following text at a comfortable speed into the area below and then
press Enter:</b> \n"+myString+"\n <b> « ENTER »</b>";
var dataString:String = ""
```

```
import flash.events.KeyboardEvent;
stage.addEventListener(KeyboardEvent.KEY_DOWN,keyDownListener);
stage.addEventListener(KeyboardEvent.KEY_UP,keyUpListener);
```

```
import flash.net.URLLoader;
import flash.net.URLLoaderDataFormat;
import flash.net.URLRequest;
```

```
var dtCurrent:Date = new Date();
```

```
function toSub():Boolean {
    if(Math.abs(type_area.text.length - myString.length) <= 10)
        return true;
    return false;
    /*
    var errors:int = 0;
    var i:int = 0, j:int = 0;
    var arr2:Array = type_area.text.split(' ');
    var arr1:Array = myText.text.split(' ');

    for(i=0; i<arr1.length; i++) {
        if(arr1[i] == arr2[j])
            j++;
        else {
            if(arr1[i+1] == arr2[j+1] || arr1[i+2] == arr2[j+2] || arr1[i+3] == arr2[j+3] || arr1[i+4]
== arr2[j+4]) {
                errors++;
                j++;
            } else
        }
    }
    /*
    for(i=0; i<myString.length; i++) {
        while(i<myString.length && myString.charAt(i) != type_area.text.charAt(i)) {
            i++;
            errors++;
        }
    }
    */
}
```

```

        if((errors+0.0) / myString.length < .94)
            return true;
        */
    return false;
}

function keyDownListener(e:KeyboardEvent) {
    if(e.keyCode == 13) {
        if(!subbed) {
            if(toSub()) {
                read_area.text = "Thanks!";
                submit();
            } else {
                read_area.text = "Too many errors, please try again. Thanks.";
                subbed = true;
            }
        }
    }
    var tString:String = (new Date()).getTime().toString();
    dataString = dataString + e.keyCode.toString()+" d "+tString.substring(8,13) + " / ";
}

function keyUpListener(e:KeyboardEvent) {
    var tString:String = (new Date()).getTime().toString();
    dataString = dataString + e.keyCode.toString()+" u "+tString.substring(8,13) + " / ";
}

function getFilename():String {
    return "" + Math.floor(Math.random()*1000000000) + ".txt";
}

function submit():void {
    subbed = true;
    var filename:String = getFilename();
    trace(filename);
    var header:URLRequestHeader = new URLRequestHeader("Content-type", "application/octet-
stream");
    var dataURLRequest:URLRequest = new URLRequest("dataSave.php?name="+filename);
    //http://www.lukeknepper.com/Research/
    dataURLRequest.requestHeaders.push(header);
    dataURLRequest.method = URLRequestMethod.POST;
    dataURLRequest.data = dataString;
    //navigateToURL(dataURLRequest, "_blank");
    var dataLoader:URLLoader = new URLLoader();
    dataLoader.dataFormat = URLLoaderDataFormat.TEXT;
    dataLoader.addEventListener(Event.COMPLETE, dataLoadHandler);
    dataLoader.addEventListener(IOErrorEvent.IO_ERROR, dataIOErrorHandler);

    try
    {

```

```
        dataLoader.load(dataURLRequest);
                read_area.text = "Thanks!!";
    }
    catch (err:SecurityError)
    {
        trace("security error: " + err.message);
    }
}

//PHP Loader stuff
function dataLoadHandler (evt:Event):void
{
    trace("subbed");
}

function dataIOErrorHandler (event:IOErrorEvent):void
{
    trace("error: " + event.text);
}
```



## **DataSave.php**

*This is the code for the PHP script that receives typing data from the DataGather fla Flash applet. It receives the data and the given filename and then creates a text file with that name and with that data on the server.*

```
<?php
    $fp = fopen( $_GET['name'], 'wb' );
if (isset($GLOBALS["HTTP_RAW_POST_DATA"]))
{
    // get bytearray
    $jpg = $GLOBALS["HTTP_RAW_POST_DATA"];

fwrite( $fp, $GLOBALS[ 'HTTP_RAW_POST_DATA' ] );
fclose( $fp );
    echo $jpg;
}
else {
    fwrite( $fp, "error" );
    echo("error");
}
fclose( $fp );
}
?>
```

## Research.html

*This is the HTML page on which I have put the data collection Flash applet. It is formatted to match the rest of my website. It displays a small amount of text with directions and information on the applet, and then of course displays the applet itself. It can be found at [www.lukeknepper.com/research.html](http://www.lukeknepper.com/research.html)*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Luke Knepper - Relaxed Creativity</title>
<style type="text/css">
<!--
body {
    background-color: #111111;
    background-image: url(designbg.jpg);
    background-repeat: no-repeat;
    background-position:top center;
    border:none;
    margin:0;
}

#design a:hover {
    background:url(design2.jpg);
    background-position:top left;
}
#flash a:hover {
    background:url(flash2.jpg);
    background-position:top left;
}
#film a:hover {
    background:url(film2.jpg);
    background-position:top left;
}
#music a:hover {
    background:url(music2.jpg);
    background-position:top left;
}
a:link {
    color: #FFFFFF;
    text-decoration: none;
}
a:visited {
    color: #FFFFFF;
    text-decoration: none;
}
a:hover {
    color: #FFFFFF;
```

```
        text-decoration: underline;
    }
    a:active {
        color: #FFFFFF;
        text-decoration: none;
    }
}
```

```
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
    font-size:12px;
    color: #CCCCCC;
    font-style: italic;
    font-weight: bold;
}
```

```
-->
</style></head>
```

```
<body>
<table align="center" border="0" cellpadding="0" cellspacing="0" width="500">
<tr height="133"><td height="133">&nbsp;</td><td colspan="2">&nbsp;<a
href="http://www.lukeknepper.com"> </a>&nbsp;</td></tr>
<tr><td width="22">&nbsp;</td>
<td colspan="2" valign="top">For the 2009-10 school year, I am conducting research on the accuracy of
using typing patterns to authenticate people. If you could really quickly (it just takes 30 seconds) follow the
directions in the Flash app below, it would be a huge help to me with my research. By typing in this applet,
you understand that your typing will be recorded and the data used for experimentation, but no personally
indentifiable information will be collected.. <a href="http://www.lukeknepper.com">Click here to return to
the homepage.</a> </td>
</tr>
<tr><td width="22">&nbsp;</td><td height="170" width="290" colspan="2">
    <p>
    <object width="400" height="300">
    <param name="movie" value="Research/driver.swf">
    <embed src="Research/driver.swf" width="400" height="300">
    </embed>
    </object>
    </p>
    <p>The Flash app.</p></td>
</tr>
</table>
<p>&nbsp;</p>
<p>&nbsp;</p>
<script type="text/javascript">
var gaJsHost = (("https:" == document.location.protocol) ? "https://ssl." : "http://www.");
document.write(unescape("%3Cscript src=" + gaJsHost + "google-analytics.com/ga.js'
type='text/javascript'%3E%3C/script%3E"));
</script>
<script type="text/javascript">
try {
```

```
var pageTracker = _gat._getTracker("UA-6483851-3");
pageTracker._trackPageview();
} catch(err) {}</script>
</body>
</html>
```