

Lab 9 Concepts: Program Design using **Model-View-Controller** Architecture (20 pts.)

Use the notes from this week's lecture and the examples posted in Canvas (and your textbook, if purchased) to help you complete the lab

Reminder: Save all your work to your *student network drive* today (**NOT** the *desktop* or *local C: drive*). At the completion of lab, put all files into a single folder, zip (i.e. compress) the folder, attach it to an email and send it to yourself. All labs are a good point of reference for projects and exams. This also serves as a possible verification of completing the lab if a discrepancy arises.

Before you begin, DOWNLOAD the 5 files for this Lab by downloading “**Lab9 Data Files.zip**” from Canvas **to your network drive** (not the local hard drive or desktop). Once you have the zip file saved, unzip all files **to your network drive** (not the local hard drive or desktop). Ask a TA if you need help.

- 1) Look over the English to Spanish dictionary (**engl2span.txt**). The first line in the file is a value representing the current number of entries in the dictionary.
- 2) Look over **Entry.java** (**model** class) and complete as instructed inside the file. Get it to **compile** before continuing.
- 3) Look over **IO.java** (**view** class) and complete as instructed inside the file. Get it to **compile** before continuing.
- 4) Look over **Converter.java** (another **model** class) and complete as instructed inside the file. Get it to **compile** before continuing.
- 5) Finish the **Lab9.java** class (**controller** class) which contains ONLY the main method.
(*Again, see the instructions within the file*)

Once complete, test your program to translate from English to Spanish. Use **engl2span.txt as the **command-line argument**. If it works OK, demo for your GTA before continuing...

Part 2: Required (FYI – this modification should only take 5-10 minutes!)

- 5) Save the **IO.java** class as **IO_GUI.java**
- 6) Modify **IO_GUI.java** to use **GUI I/O** instead of **Console** (text-based) I/O

Here is a refresher on how to code a GUI...

```
import javax.swing.JOptionPane;    // import for GUI interface
```

For **input**: `JOptionPane.showInputDialog("Prompt Message");` ... returns a String, like `s.nextLine()`

For **output**: `JOptionPane.showMessageDialog(null, "String Msg");` ... displays a String, like `println`

- 7) In your controller class, create an **IO_GUI** object instead of an **IO** object. Everything else should remain the same. This illustrates the power of MVC architecture. You can change the **view** without changing the **model**!

Once complete, test your **GUI-based program to translate from English to Spanish. Use **engl2span.txt** as the **command-line argument**. If it works OK, demo for your GTA and you are done!

After all parts are working correctly, **demonstrate your working program for your GTA**. Don't leave until the GTA has seen your program run, checked you off the list and you have uploaded your **files** to Canvas.

For this lab, **place ALL java files** (you should have 5) in a folder called **Lab9**. **Zip that folder** and submit to Canvas (only a .zip will be accepted in Canvas). **Verify that the zip file has been properly submitted before leaving!** It is recommended that you attach your file(s) to an email and send it to yourself. All labs are a good point of reference for projects and exams. This also serves as a possible verification of completing the lab if a discrepancy arises.