

Lab Exercise #7  
*Model, View, Controller Paradigm*  
*Computer Science 2334*  
**Due by: Friday, April 10, 2009, 2:00 pm**

Members:

---

---

---

---

---

**Learning Objectives:**

- Understand how to create a model in the MVC paradigm.
- Understand how to create a view in the MVC paradigm.
- Understand how to create a controller in the MVC paradigm.
- Understand how models and views are connected to one another and communicate with one another in the MVC paradigm.
- Understand how views and controllers are connected to one another and communicate with one another in the MVC paradigm.
- Understand how controllers and models are connected to one another and communicate with one another in the MVC paradigm.

In this Lab you will implement a simple graphical display system for contact information. The display system allows a user to select from among a list of contact names and thereby add full contact information to a display list. For this exercise, you will complete the model, the views, and the controller for this system and hook them together into a working system.

**Instructions:**

This lab exercise requires a laptop with an Internet connection. Once you have completed the exercises in this document, your group will submit it for grading. All group members should legibly write their names at the top of this lab handout.

***Make sure you read this handout and look at all of the source code posted on the class website for this lab exercise before you begin working.***

1. Review the source code posted on the class website. Pay close attention to the MVC classes, which are the **Model**, **MasterListView**, **SubListView**, and **Controller** classes.
2. Read through the source code of the **Model** class and note the comments provided in the source code that give hints as to what needs to be done in the program.
3. Declare variables called **addressBook** and **subList** to hold the information in the model for the application objects being modeled. These variables must be of appropriate types to hold lists of contact information. (See the **Contact** class for details on what information is available for each contact.)

What data structures will be used for `addressBook` and `subList` and why?

4. Declare a variable called `actionListenerList` to hold the information in the model that allows it to be the source for action events. This variable must be of an appropriate type to hold a list of listeners.

What data structure will be used for `actionListenerList` and why?

5. Complete the `model` class by adding a call to its `processEvent()` method. This method should be called with an `actionEvent` that has `"subList"` as its command string.

Where should `processEvent()` be called? Why?

6. Read through the source code of the `MasterListView` class and note the comments provided in the source code that give hints as to what needs to be done in the program.

7. Complete the `setModel()` method for `MasterListView` by registering the view.

Will the `MasterListView` object be a source or a listener in this part of the code? Why?

To which class will the object at the other end of this source/listener connection belong? Why? Will that other object be a source or a listener? Why?

8. Read through the source code of the `SubListView` class and note the comments provided in the source code that give hints as to what needs to be done in the program.

9. Complete the `setModel()` method for `SubListView` by registering the view.

Will the `SubListView` object be a source or a listener in this part of the code? Why?

To which class will the object at the other end of this source/listener connection belong? Why? Will that other object be a source or a listener? Why?

10. Read through the source code of the **Controller** class and note the comments provided in the source code that give hints as to what needs to be done in the program.

11. Complete the **setView()** method for **Controller** by registering the controller.

Will the **Controller** object be a source or a listener in this part of the code? Why?

To which class will the object at the other end of this source/listener connection belong? Why? Will that other object be a source or a listener? Why?

12. Test the program to make sure it correctly responds to list selection events.

13. Submit the project archive following the steps given in the Submission Instructions by 2:00 pm, Friday, April 10, 2009.

14. Turn in this lab handout to your lab instructor or one of the other instructors by 2:00 pm, Friday, April 10, 2009.