CPSC 2151

Lab 8 – Introduction to MVC

Assigned: Tuesday 11-7 && Thursday 11-9

Due: Tuesday 10-28 && Thursday 10-29 (**Before lab starts**)

In this lab, you will be working on a small project that should help visualize one of the many ways we often use the Model-View-Controller Architectural Pattern.

You'll start out with the starter code provided on Canvas. The starter code contains 4 files:

- Lab8.ClassController.java – The controller of the MVC, used to mediate the communication between the model and the view.
- Lab8.Models.ClassModel.java – A model that represents a Clemson course listing.
- Lab8.Views.ClassView.java – A "view" that serves as a frontend for our course listing program.
    - Note that since we haven't covered GUIs yet, our "view" isn't a *real* view, and the best we can do is print to the console/terminal.
- Lab8.MVCMain.java – A main method file. Mostly used for verifying your solution.

Your job for this lab is to complete the following:

1. Read the contracts for each of the functions contained in the Model, View, and Controller classes.
    a. The goal is to gain a **basic** understanding of what each of the three types of files do to support the project.
    b. As you look through the contracts (and eventually implement the functions), try to remember some of things we have discussed in class:
        i. Why is it important that the models and the views don't interact directly with each other?
        ii. MVC is most often used for front-end development, which means we're working with end-users to some degree. Where is it appropriate to do the in-method error checking we've been able to avoid with just our contracts?
        iii. How does the controller serve as a "mediator" between the models and the views?
2. Implement the functions based on the contracts.

This is a very short lab. If we had more time in class, this would have been an in-class demo/example instead of a full lab... Until we talk about GUIs, there's only so much that we can do to mimic the real usage of MVC. Worry not, for lab 9 will continue this lab after we talk about GUIs and will provide plenty more work for you to do :)

Partners:

I personally recommend you work individually for this lab since it's so small and it's less about coding and more about getting hands-on exposure to one of the many ways we can use MVC. However, you're welcome to work individually or with your normal teams of 3-4 total students.

Understand that your 2nd lecture exam, the final (especially the final), and the GUI unit will cover MVC extensively, and if you don't understand what a controller does because you worked in a group and divided-and-conquered the lab, resulting in you only getting exposed to working with the make-shift view, that's on you…

Submission:

When you go to submit, you are submitting the individual model, view, and controller files:

- ClassModel.java
- ClassView.java
- ClassController.java

Even though you are submitting these three files **individually**, not the entire project folder…

DO NOT CHANGE THE PACKAGE STATEMENTS FOUND ON LINE 1 OF EACH OF THESE FILES.

The autograder will look for just these three individual files. If you submit the entire project folder, or these three files as a zip, or anything other than the three individual .java files, the autograder won't know where to look for your submission…

If you change the package statements from what they currently are, when the tests go to run Java won't be able to compile the three files since a mismatch of package structure will throw an error.

The autograder is 100% of the grade for this lab, and it's all there.