**PUI Prototype Lab Homework 1: Flex Warm-up**

# Objective

This goal of this assignment is to demonstrate that you can use Adobe Flex by creating three small programs. This will require going through a simple tutorial and then playing with the system and reading more on your own about Flex. Warning! This should be an easy assignment if you start early. It's ugly if you don't. Students who do not complete this assignment on time will be assumed to not have the pre-requisites for the course and will not be allowed to continue in the course.

# Resources

You must use Adobe Flex with ActionScript 3 to do this assignment. You should have downloaded and set up Flash Builder 4.6 (DO NOT USE 4.7, see [course website](http://www.andrew.cmu.edu/user/ihowley/pui/index.htm)) by now, which you will use to complete this assignment.

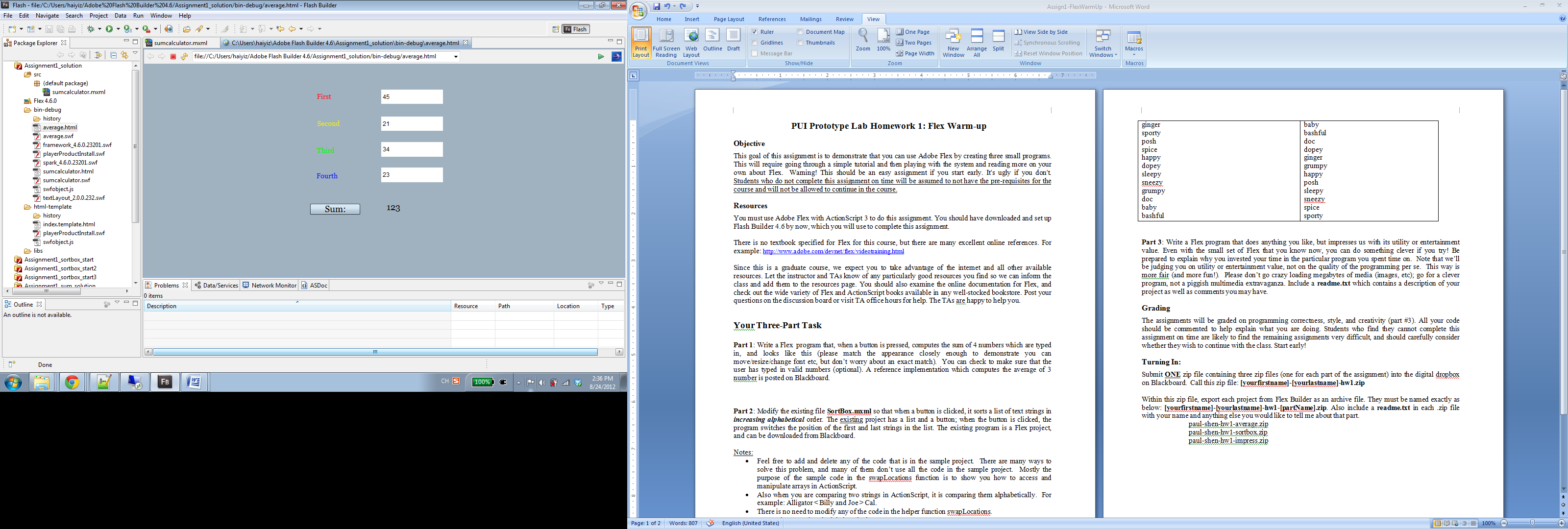
There is no textbook specified for Flex for this course, but there are many excellent online references. For example: <http://www.adobe.com/devnet/flex/videotraining.html>

Since this is a graduate course, we expect you to take advantage of the internet and all other available resources. Let the instructor and TAs know of any particularly good resources you find so we can inform the class and add them to the resources page. You should also examine the online documentation for Flex, and check out the wide variety of Flex and ActionScript books available in any well-stocked bookstore. Post your questions on the discussion board or visit TA office hours for help. The TAs are happy to help you.

To open reference implementations in FlashBuilder 4.6 go to ‘File > Import FlashBuilder Project’.

**Your Three-Part Task**

**Part 1**: Write a Flex program that, when a button is pressed, computes the sum of 4 numbers which are typed in, and looks like this (please match the appearance closely enough to demonstrate you can move/resize/change font etc, but don’t worry about an exact match). You can check to make sure that the user has typed in valid numbers (optional). A reference implementation, hw1-sum-start, which computes the average of 3 numbers is posted on the course website.



**Part 2**: Modify the existing file **hw1-sortbox-start.mxml** so that when a button is clicked, it sorts a list of text strings in ***decreasing alphabetical*** order. The existing project has a list and a button; when the button is clicked, the program switches the position of the first and last strings in the list. The existing program is a Flex project, and can be downloaded from the course website.

Notes:

* Try to use the swapLocations function in your implementation. (Hint: a ‘selection sort’ is one of the simplest sorts to implement).
* Feel free to add and delete any of the code that is in the sample project. There are many ways to solve this problem, and many of them don’t use all the code in the sample project. Mostly the purpose of the sample code in the swapLocations function is to show you how to access and manipulate arrays in ActionScript.
* Also when you are comparing two strings in ActionScript, it is comparing them alphabetically. For example: Alligator < Billy and Joe > Cal.
* There is no need to modify any of the code in the helper function swapLocations.
* Remember to update the label on the button.

**Be sure to add comments that explain how you are sorting the strings.**

|  |  |
| --- | --- |
| You want your final code to sort the following list: | So it is in alphabetical order (like this) |
| ginger  sporty  posh  spice  happy  dopey  sleepy  sneezy  grumpy  doc  baby  bashful | sporty  spice  sneezy  sleepy  posh  happy  grumpy  ginger  dopey  doc  bashful  baby |

**Part 3**: Write a Flex program that does anything you like, but impresses us with its utility or entertainment value. Even with the small set of Flex that you know now, you can do something clever if you try! Be prepared to explain why you invested your time in the particular program you spent time on. Note that we’ll be judging you on utility or entertainment value, not on the quality of the programming per se. This way is more fair (and more fun!). Please don’t go crazy loading megabytes of media (images, etc); go for a clever program, not a piggish multimedia extravaganza. Include a **readme.txt** which contains a description of your project as well as comments you may have.

# Grading

The assignments will be graded on programming correctness, style, and creativity (part #3). All your code should be commented to help explain what you are doing. Students who find they cannot complete this assignment on time are likely to find the remaining assignments very difficult, and should carefully consider whether they wish to continue with the class. Start early!

# Turning In:

Submit **ONE** zip file containing three fxp files (one for each part of the assignment) to your TA via email.

Call the zip file: **[yourfirstname]-[yourlastname]-hw1.zip**

Your e-mail subject should be: **PUI HW1 [yourlastname]**

Within this zip file, export each project from Flex Builder as a flash builder project. They must be named exactly as below: **[yourfirstname]-[yourlastname]-hw1-[partName].fxp**. Also include a **readme.txt** in each .zip file with your name and anything else you would like to tell me about the assignment.

iris-howley-hw1-sum.fxp

iris-howley -hw1-sortbox.fxp

iris-howley -hw1-impress.fxp

readme.txt