Information Infrastructure II

INFO I211 Spring 2014

Final Exam *Practice* 2

Instructions

Perform the requested actions for the following task.

Submit the solution to the programming exercise code to Oncourse under "20140501-02 Lab 15 task".

Also make sure you click through to the final "Submit" button, as with other Oncourse submissions. You are responsible for making sure that your work is submitted by the end of the lab where you are taking the final exam: it is strongly advised that you verify the submission before leaving the lab. Late submissions can not be accepted.

As a reminder, during the exam you may consult anything on the Oncourse section for this class, as well as your textbook (the Python textbook, by Michael Dawson). You may not bring in any additional printouts, nor may you use the internet except to access Oncourse and the electronic version of the textbook.

I211: Information Infrastructure II - Final Exam *Practice* 2

Submit your program to Oncourse under "20140501-02 Lab 15 task". **Make sure to submit the correct file(s) to the correct OnCourse assignment!**

Make sure you click through to the final "Submit" button, as you must with homework assignment submissions. You are responsible for making sure your work is submitted **by the end of the lab**, so it is strongly advised that you verify the submission before leaving the lab. **We can NOT accept late submissions.**

As a reminder, you may use anything on the Oncourse section for THIS CLASS, as well as your textbook. You may not bring in any additional printouts, nor may you use the internet except to access Oncourse.

CGI input processing and Javascript input processing:

The task for Task 2, Problem A, is to complete the "Name Entry Program". The task consists of 2 parts. Both parts *need to reside on the server system running CGI scripts and an HTTP server*, i.e. silo.soic.indiana.edu, in your ~/cgi-pub/ directory.

The two parts are as follows:

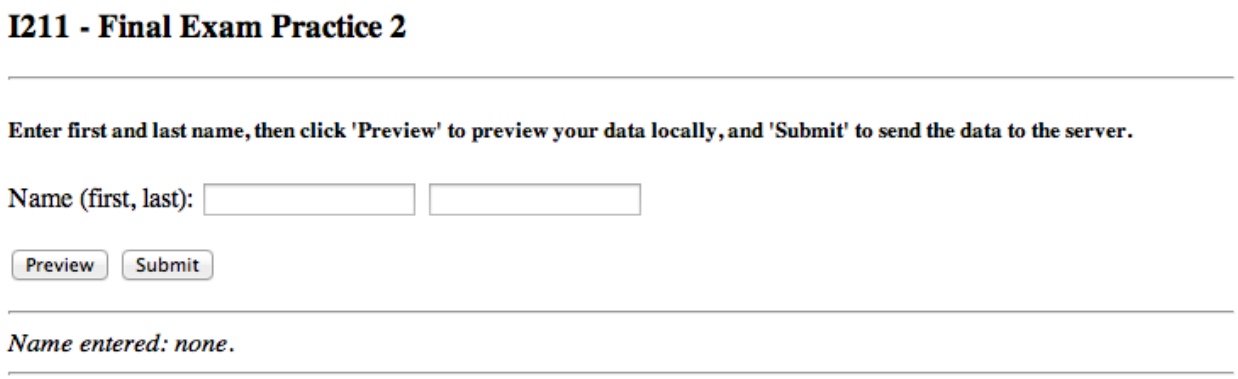
1. an HTML page "i211finalPractice2.html" to collect input from users online by using <form> and/or <input> elements, which will include text boxes, and two buttons to submit the data. In Javascript, write a script to be included in the "i211finalPractice2.html" page, to present a preview of the provided input for first and last name, *before* it is sent to the CGI server with the submit button.

2. a Python 2.x CGI script "i211finalPractice2.cgi" to collect the provided input about first and last name, save (append, not overwrite!) the data to a file named "names.txt" in the same ~/cgi-pub/ directory where the html and cgi files reside, and dynamically generate a web page to display all entered names.

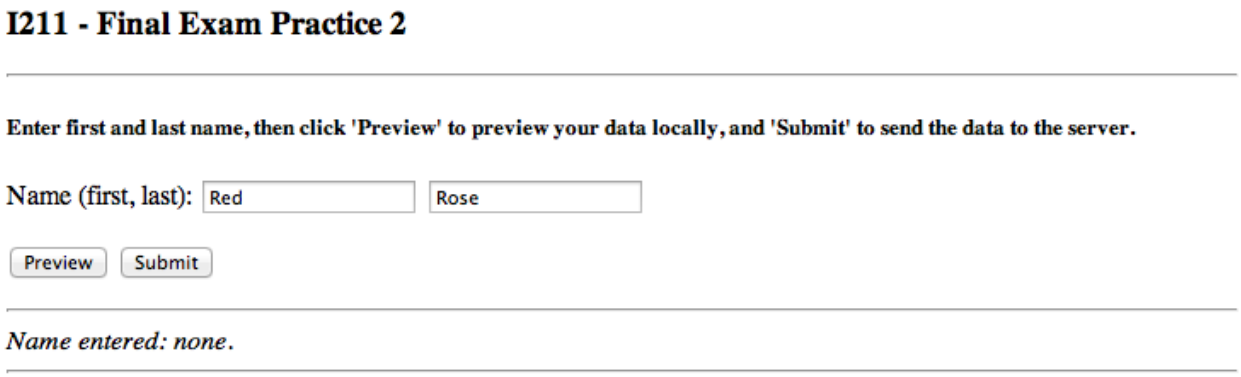
In Python 2.x on silo.soic.indiana.edu, write a *CGI script* called "i211finalPractice2.cgi" that implements the following program for ordering names.

A starting text for the HTML page is provided to you in the file "i211finalPractice2.html" available on Oncourse.

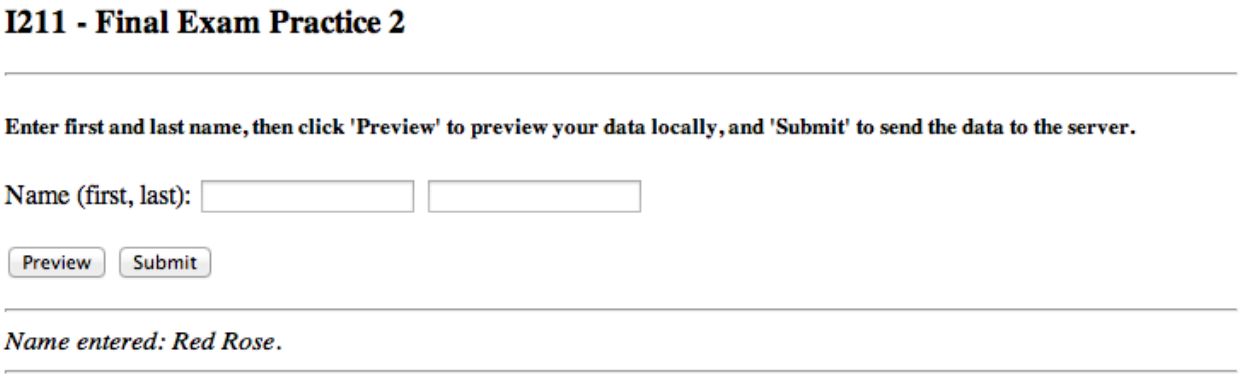
The HTML page should look like this when opening it in a browser:



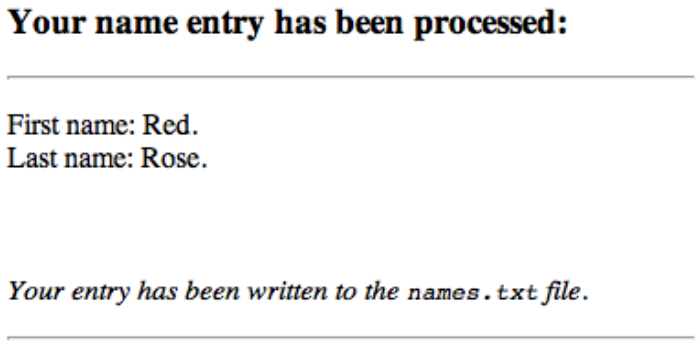
After entering information for first and last name, and before clicking any button, the web page should display as follows:



After clicking “Preview”, the page should display as follows, with the bottom line generated by your Javascript code:



And after clicking "Submit", the resulting web page as generated by your "i211finalPractice2.cgi" script should display as follows:



After having submitted three entries for names, e.g. Red Rose, Jack Spratt and Thursday Next, there should be a text file in the same directory where your CGI and HTML files are located, called "names.txt" with this information in it:

```
First name: Red.
Last name: Rose.
------------------------------
First name: Jack.
Last name: Spratt.
------------------------------
First name: Thursday.
Last name: Next.
------------------------------
```

Other program requirements are: an entry can be placed without either first or last name, but at least one of them needs to be provided. If the user does not enter a first name, the first name should be recorded as 'unspecified'; likewise for a missing last name. If there is an error in writing the file "names.txt", the resulting web page should display the following notice at the bottom of the page: *An error occured while writing to the names.txt file.*

This task is worth 70 points:

- You can get up to 40 points for writing a program that simply works, regardless of how it does so: 20 points for the CGI Python script, and 20 points for the Javascript code.
- To get up to 5 more points, you must wrap up your Python code into an Application class.
- To get up to 10 more points, organize your programs into logical functions. Use a separate function for each major portion of the program.
- To get 10 more points, comment your code well.
- To get the final 5 points, you should implement validation and error handling on anything where problems might occur in your code.

Poor coding habits, lack of comments, and other minor errors will subtract points. Do not use global variables. Do not shadow variables.

Make sure to add comments on top of your Python 2.x code and your HTML file that identifies the lab in which you took this exam, your name, and your IU username!

Good luck!