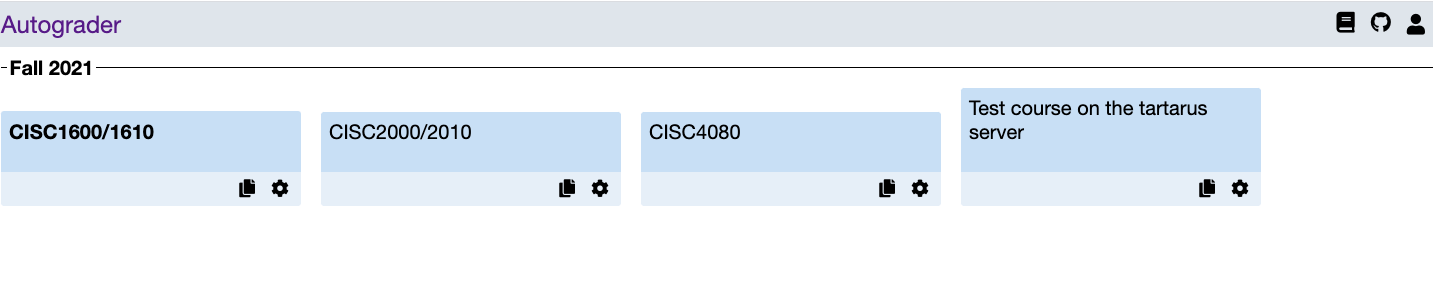
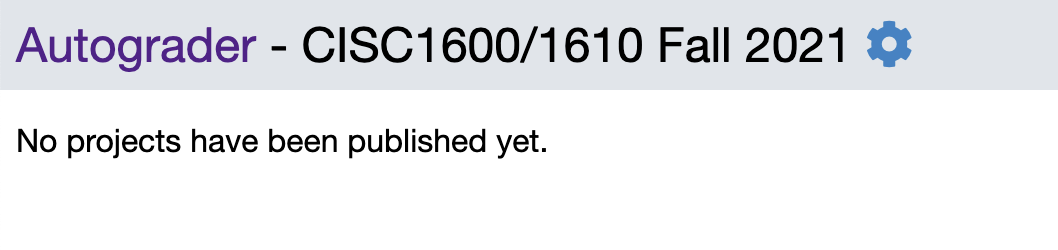
**Week #1: July 14, 2021 - July 2021, 2021**

**Practice log on to the testing system, and explore adding a project into the course.**

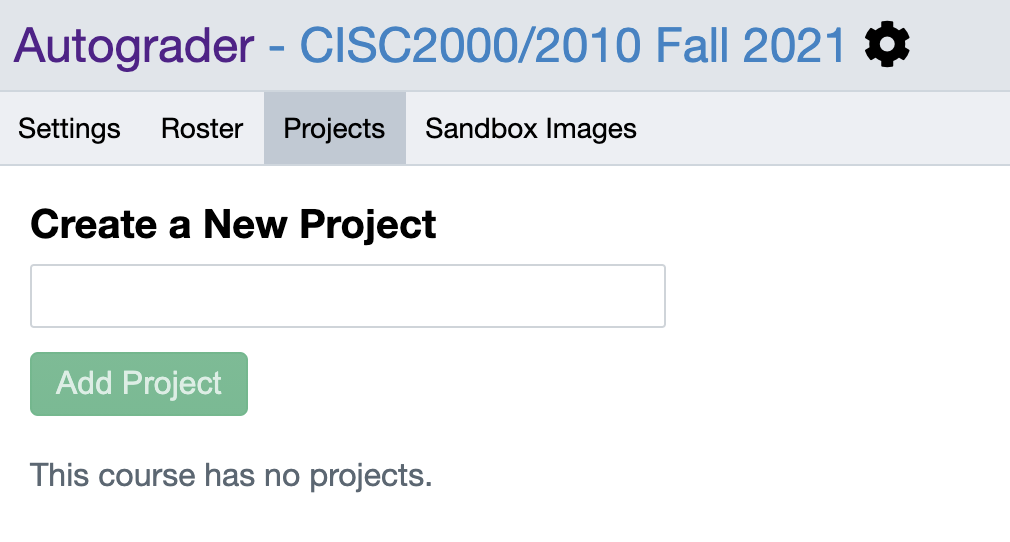
* **The autograder system has been installed and you can access it at:** [**https://tartarus.cis.fordham.edu:8443/**](https://tartarus.cis.fordham.edu:8443/)
* **You have been added as “staff”and/or admin user to the three courses below (CS1600/1610, CS2000/2010 and CISC4080). All three courses are empty.**
* **You can click on the first button to the right of the header line to access the manual, which takes you to: https://eecs-autograder.github.io/autograder.io/**

****

* Click on the course that’s assigned to you, you will see a page like belows:



* Click on the blue button/icon next to the course name, you will see a page as below, from which you can add user, add project, and so on:



Winsor Tse:[wtse@fordham.edu](mailto:wtse@fordham.edu)>:

Week #1 Tasks:

* Add the following project to CISC2000/2010, Palindrome (see the folder for the instructions and test cases)
* When you are done, add Maria as a student to the class for her to submit a solution to the program and see the grading result.

Week #2 Tasks:

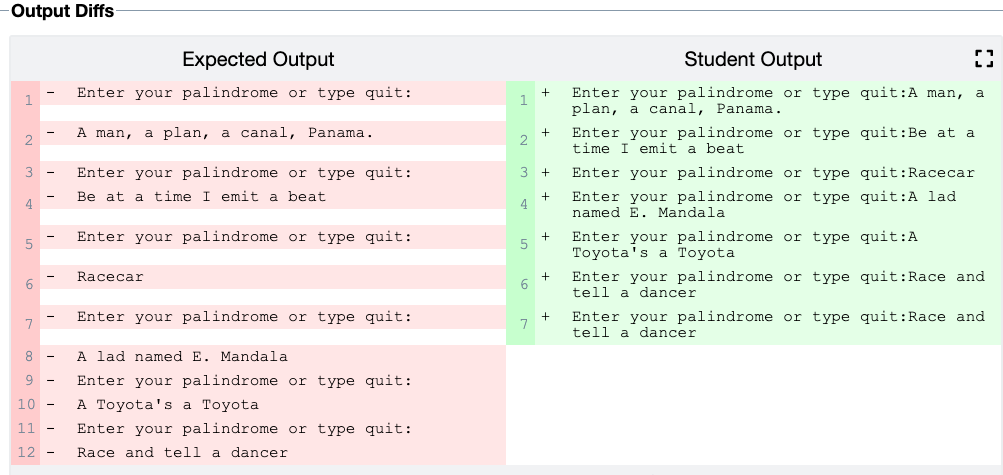
* The current test cases show to the student the expected output. Look into how to add test cases that are hidden from students (i.e., students can only see that they pass or fail the test cases, but cannot see the expected output.)
* Look into:
  + Design multiple test cases for this project, and assign a certain number of points for each test case.
  + How to add coding style checking, memory leak checking or running time checking as aspects of grading? (I added a simple testing where I checked the # of lines in source code. We are looking into installing some tools to the server which we can use to check style.)
    - For now, familiarize yourself with <https://oclint.org/>
    - Modify the CodingStyle test case so that a program passes this test case if the program has less than 100 lines of code. (Note: consider using bash scripting).

Week #3:

* Please import the following lab into CISC2000 course and set up test cases:

The lab description is given in directory: <https://drive.google.com/drive/folders/1fYZmktk8YAATWsaUs-mBFRAsaYePcADQ?usp=sharing>

* The lab description, solution code, and all test cases are given.
* When you are done, practice submitting some codes (correct or wrong ones) and see if the testcases run correctly.



Maria Jara <[mjara2@fordham.edu](mailto:mjara2@fordham.edu)>:

* Add the following project to CISC1600/1610: Lab03 Fordham Air <https://drive.google.com/file/d/1wf-jFjeOZABd9UmpDc-20nJow-qYF_JY/view?usp=sharing>
* Add a few test cases for autograding
* When you are done, add Winsor as a student to the class for him to submit a solution to the program and see the grading result.

Feedback/Comments, Week #2:

* I added you as admin to CISC2000/2010 and CISC4080, and you can see how Winsor and Coutney set up these test cases. The test suite should be used to specify
  + How to build/compile the program, before we can run it for testing purposes
  + We can add multiple test cases in each suite (see the CISC2000/2010 for example).
* Finish importing the FordhamAir project.

Courtney King <[cking74@fordham.edu](mailto:cking74@fordham.edu)>

Week #1:

* Add CISC4080 Lab01 to CISC4080
* Explore set up unit testing cases for bubble sort, selection sort, and binary search.
* When you are done, add Winsor and me ([xzhang@fordham.edu](mailto:xzhang@fordham.edu)) as students to the class.

Feedback/Comment: Week #2:

* Good job figuring out unit testing!
* Explore an alternative way to perform unit testing: where multiple test cases are coded in the program, [as illustrated in this file](https://storm.cis.fordham.edu/~zhang/cs4080/Demo/Lab3/lab3_test.cpp), and we return 1/0 for failure/pass. (See the code for detailed comments)
  + Convert this given file to use for lab1
* Look into how to measure running time?
  + Can be done in test driver code.
* Look into how to download grades for students. Is there a gradebook functionality available?
  + Yes.
* Is there any way to check code similarity across different students’ codes? : no apparently…

Week 3:

* Please import the following lab into CISC2000 course and set up test cases:

The lab description is given in directory: https://drive.google.com/drive/folders/1tsBXmMF3x3O\_REnkqtW2lL\_1CnedbwMc?usp=sharing

* The lab description, all test cases, and solution code are included in the folder.
* When you are done, practice submitting some codes (correct or wrong ones) and see if the test cases run correctly. You can introduce some errors to the correct code for testing purposes.

From Courtney: I didn't have the option to add the project to CISC2000 so it’s currently in CISC4080. For the implementation, I wrote a .cpp file to use the assert code given in 5 of the test cases and textbox i/o for the other 3. I submitted the given solution code which passes all the tests, and a few other submissions with various mistakes, which are commented in the submission files. I’m not 100% sure if I correctly interpreted the tests given or the preferred implementation, so please correct me if anything is off.

**Please feel free to email me if you have any questions. Let’s plan to have a meeting early next week for questions/updates.**