

PIC 16, Fall 2016 – Final Exam

Administered 3:00 p.m. – 6:00 p.m. Wednesday, December 7, 2016. You may reference any of your work from the course and any documents I have provided. Links to other allowed resources are provided in the exam package (to which I will provide a link shortly). If a resource you feel you need is not listed, you may ask about it during the exam.

There are two parts to the exam. Part I is required and Part II is optional (with one exception... read on). Part I tests Track A/B material from the latter part of the course. For this part, you will get to choose one of six options (three corresponding with each track). Typically, these prompts ask you to complete one relatively long, multi-part objective.

If you are not confident about your performance on Part I, you may complete one or more options from Part II, which tests common material from the first part of the course. Typically, these options are shorter than those in Part I or are composed of several unrelated, very short questions.

All but one of these options will require the use of Jupyter Notebook. You are allowed to prepare your solutions in whatever program you wish, but you are required to submit your solution in the provided Jupyter Notebook.

Part I – Track B – Option 3 is the only oddball. You submit an entire Scrappy project folder. Also, if you choose this option for Part I, you must complete one (your choice) option from Part II.

A weighted average of your scores for Part I and Part II will be used to calculate your final exam score.

For now, log on to your computer. If you still don't know where Jupyter can load from and save to, right click the Jupyter icon on the desktop. The "Start in" folder is the root of the folders that Jupyter can access. If it is not the root of your (personal) Z:\ drive, I suggest you change the value in the box to "Z:\:" before starting Jupyter. The reason is that Z:\ is very easy to access from the Windows file explorer, so it will be easy to place the template exam notebooks there and upload your solutions from there when you're done. Plus, Z:\ is a network drive, which is safer and more accessible (to you) than your machine.

Next, consider which of the following Part I exam options you might like to pursue. Summaries are provided here; detailed task descriptions will be provided in the exam package. You are required to submit a solution to *one only*, and only one solution will be graded per person.

Track A

Option 1) Ballistic Projectile Trajectory Optimization: In this exam option you will use `scipy.integrate.odeint` to simulate the motion of a ballistic projectile, that is, an object that moves under the influence of gravity alone. Then you will use your simulation in conjunction with `scipy.optimize.minimize` in order to find the initial velocity (speed and direction) that will cause your projectile to hit a desired target with minimum initial kinetic energy (speed). I think this option requires the least code, and it's a cinch if you know what you're doing. But you have to know what you're doing.

Option 2) Musical Instrument Tone Classification: Your objective in this exam option is to train a classifier to distinguish among musical instruments playing the same note. This is possible because in addition to the "fundamental" frequency, musical instruments also resonate in characteristic amounts at integer multiples of the fundamental called "harmonics". After taking the FFT with `scipy.fftpack.fft`, you will extract the power of each harmonic relative to the power of the fundamental, and use these as features to train (and test) an `sklearn.svm.LinearSVC`. For me, this was definitely the most interesting Part A option and likely the most challenging. That's probably because I wrote code like Option 1 (in my sleep and again for breakfast) throughout my PhD.

Option 3) Image Processing Two Ways: We discussed in class that OpenCV comes with some handy functions, but that many of these you could write yourself with a few lines of Numpy. In this exam option, you'll show me that you can use the OpenCV functions and also that you can write them yourself. In my opinion, this is the least interesting and longest but most straightforward of the Track A options. To do well, you need some understanding of OpenCV and decent understanding of Numpy.

Track B

Option 1) Random Number Generator: This is my personal favorite. In one of the preparations we noted that we cannot count on the operating system to switch among threads in a predictable fashion and that we might be able to use this idea to create a random number generator. In this exam option, you will follow up on this idea. The task is to create what I'll call an "arbitrary" number generator, because the numbers generated are not truly random nor do they follow a familiar distribution like those from a pseudorandom number generator. This can be pretty short, and it should be quite manageable if you can make a basic Plotly plot and really know how `threading.Thread`s work. It also rivals Track A Option 2 for how interesting it is. Stats majors should love it (but the only stats you need to understand are averages and histograms).

Option 2) Random Haiku Generator: A haiku is defined not by a rhyme pattern, but by the number of syllables in each line. Traditionally, a haiku has three lines: 5 syllables in the first line, 7 in the second, and 5 in the third. In this option, you will write a program that will generate haikus worthy of literary praise (almost surely). Of course, it will generate many, many more bad haikus, like:

gnatcatcher julep
renewable unite male
miscreation loll

This involves some new (but very simple) features of NLTK: a dictionary of word pronunciations and a separate list of English words. Besides the Plotly, I think I'll make this an exercise for 10B this spring.

Option 3) Whoa... meta: In this option, you'll use Scrapy to scrape data about data. Specifically, you'll write a spider that can download the titles, descriptions, provider organizations, and contact information for all the U.S. Government's open data available at `data.gov`. The spider will look exactly like all the other spiders we've written in this class... except that the website and thus all the CSS expressions are totally different, and those can be tricky to debug. *Since this option requires so little original Python code, you are required to complete one (your choice) of the Part II options if you choose this for Part I. If you submit only Part I Track B Option 3 without any Part II options, the maximum score is 70%.*

If you're not comfortable with your performance on Part I, there is an optional Part II. You may complete *one or more* of three options for Part II. If you do so, your score for the options you complete will be factored into your total score for the exam with the weights below. (The weight of Part I will be reduced accordingly.)

- Option 1: Ten separate, short questions on material from the first few weeks. Weight is 20%.
- Option 2: Two medium-length questions on material from the first few weeks. Weight is 20%.
- Option 3: A single, multi-part GUI/animation problem. Weight is 30%.

For instance, let's say you were to score 70 points out of 100 on Part I. You also attempted Part II, Option 1, scoring 35 out of the 40 points available for that option. Finally, you somehow had time to complete Option 3, scoring 20 out of the 20 points available for that option. Then your grade for the exam would be: $70/100 \times 50\% + 35/40 \times 20\% + 20/20 \times 30\% = 82.5\%$

When you're done, submit on CCLE a `.zip` containing *only* your solution files. Good luck!