Name: _____

PIC 97, Winter 2016 – Final Exam

Administered 11:00 a.m. – 11:50 a.m. Friday, March 11, 2016. Corresponds with 8M-10M.
No outside resources other than features of Python software (Spyder, IPython, etc…), official Python, NumPy, SciPy, matplotlib, OpenCV, and Scrapy documentation (found on their respective official websites) and the provided links (in the exam assignment page on CCLE) are allowed. **You may use a search engine to find these pages,** but visiting other websites (including those linked from the allowed websites, unless otherwise specified) is not acceptable.

Flip this sheet only when instructed.

(Until then, squint as much as you wish trying to read backwards through the page.)

The exam questions are contained in this document, but you are to submit your solution in a Python module `final.py`. When you are done with the exam, submit `final.py` and any other files (e.g. a `.csv` or `.avi` file you produce, etc…) in a `.zip` file `final.zip` on CCLE. Of course, you are welcome to write code in any software installed on the PIC machines and copy it into the requested files when you're done. In fact, I recommend saving backups of all your code in a simple text document `backup.txt` and including that in your `.zip` file as well in case the primary files do not save properly.

For now, log on to your computer, start Spyder, and get familiar with the allowed resources.

You are only required to submit a solution to *one* of the three problems within. If you submit solutions to multiple problems, only one will be graded. I consider them to be of approximately equal difficulty and length. I suggest that you choose the one that looks easiest to you.

The references linked from CCLE for the first and second midterm are still allowed and links are still available on CCLE. References more relevant to this exam are:

SciPy Lectures (http://www.scipy-lectures.org/intro/index.html)
OpenCV Tutorials (http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html)
Scrapy Tutorial (http://doc.scrapy.org/en/1.0/intro/tutorial.html)

Good luck!

Name: _____

1. SciPy
    a. (45pt) Solve the Van der Pol equation[1] with $\mu = 0.1$ and initial conditions $x(0) = 0.1$, $\dot{x}(0) = 0.1$ for $0 \leq t \leq 100$. Plot $x(t)$ *against* $\dot{x}(t)$ (that is, the horizontal axis is for x and the vertical axis is for $\dot{x}$ on the same plot. This is called a phase plot, and in this case it makes sense to plot these variables against one another).
    b. Consider the Rosenbrock function[2] with $a = 1$ and $b = 100$.
        i. (40pt) Find the minimum using SciPy
        ii. (5pt) Find the minimum within the bounds $0 \leq x \leq 0.75$, $0 \leq y \leq 0.75$
        iii. (10pt) Find the minimum within the circle of radius $r = 0.25$ centered at the point $(0.25, 0.25)$. For reference, the equation for a circle with radius r centered at (x_c, y_c) is $(x - x_c)^2 + (y - y_c)^2 = r^2$.

2. OpenCV (You were responsible for ensuring that OpenCV works on your computer. Whether `ffmpeg` is set up properly or not, you can still write correct code.)
    a. (50pt) Load the image `a.png`[3] and save a two-second video `a.avi` from it. (Every frame is `a.png`.) Be careful to specify the frame dimensions correctly in the constructor of the object that writes video. It is expecting a tuple like (*width*, *height*). If you provide the dimensions incorrectly, you will not get an error, nor will you get a valid video file.
    b. (30pt) Play the two-second long "movie" of `a.png` in a window that can be closed by pressing "q" on the keyboard.
    c. (5pt) Save a video four seconds long in which the first two seconds are `a.png` and the second two seconds are `b.png`[4].
    d. (15pt) Save a video like one of the above, except that the video fades to black in the last second. A concept for darkening pixels (or an entire frame!) is represented by the equation:
    $$f(\rho) = f_0(1 - \rho) + b\rho$$
    where f_0 is the original pixel color (or frame), b is pure black (or a pure black frame), and ρ is a parameter you vary from 0 to 1. When ρ is 0, the color/frame is unchanged; when ρ is 1, the color/frame is black; when ρ is 0.5, the color/frame is half-way in between the original and all black; etc... Note that performing such math may automatically promote the data to a different data type. You may need to manually change the data back to the original type

3. Scrapy (Choose this at your own risk. If craigslist starts to block your spider, you'll have difficulty testing your code, but it's still possible to write it!)
    a. (75pt) Scrape the titles of the listings on the first page of craigslist computer gigs in Los Angeles[5] and save them in a `.csv` file
    b. (10 pt) Scrape this information from the first *N* pages of listings and save them in a `.csv` file. You should have a variable `n_pages` in your code, and simply changing the value of the variable should change the number of pages scraped. Use `n_pages = 2`.
    c. (10pt) Scrape the links, titles, and descriptions (from within each listing) and save them in a `.csv` file. Note that you may find it easier to change the way you scrape the titles...
    d. (5 pt) Ensure that you have saved the *whole* description, not just the part before the first `<br>` tag. Replace any line break characters with single spaces.

---

[1] https://en.wikipedia.org/wiki/Van_der_Pol_oscillator
[2] https://en.wikipedia.org/wiki/Rosenbrock_function
[3] https://www.dropbox.com/s/85gnehrtipbpn1u/a.png?dl=0
[4] https://www.dropbox.com/s/8wsmzvpmejf1ezg/b.png?dl=0
[5] http://losangeles.craigslist.org/search/cpg