Project 2

Due: May 24th, 2015

You are required to write a Python program that clearly shows me you are proficient in Python. The purpose or use of the program is completely open.

Though there are no other explicit requirements, if you are unsure as to whether your project will show a genuine understanding of Python, you can considering the following guidelines.

- Use classes and their methods extensively

- Include use of file I/O and/or 3rd party modules

- Do not simply rehash what we did in class. Take the concepts we've learned and make them do something new

- Use a concept or 3rd party module we didn't cover in class

- Make it unlikely that the user will run into bugs or errors (i.e. check for bad input in programs that accept input)

- Expect to write **at least** 400 lines of code. If half your code is print statements, expect on writing more. If your code uses a lot of clever recursion, you might be able to write less.

Again, these are only suggestions for those who don't know if their program will show their proficiency in Python. If you have a great idea for your program and feel confident that it clearly shows you are skilled in Python, you are not required to do all of the above. If you still have no idea what to work on, contact me and I can help you decide on an idea.

Group Work: You may work in a group only if you use a version control system, such as Git. I will expect to see the version control repository during submission. I expect to see several different alternating commits by each student (only having one huge

commit from each student at the end of the semester is a sign that one student did all the work and then just committed her own code for the other student). I require the use of version control so that it is clear that every student did an enough work.

Extensions: The project is due the last day of finals and the required administrative grade submissions are due shortly after this. Because of this, there will be no deadline extensions on this project, excluding cases such as medical emergencies. Complete your project early to insure you don't run into issues. **Back up your code online.** Computers breaking and accidentally deleted or lost code will not be accepted as an excuse.

Plagiarism: Claiming code from resources online will not be tolerated and will give you a zero on the project. You're much better off turning in a halfway done project than changing the names of variables in someone else's code. If you use code from others online or elsewhere, seperate it from your own code and clearly mark it as someone else's during submission.

Extending Project 1: If you plan to extend your project 1 code, you have to add as much extra code to your project 1 as you would have had to write if you were starting a new project 2 (I will take into account the fact that larger projects are harder to work with, so you'll actually need to write slightly less).