**ITP 125 – Programming Project with Python**

**(Instructor Chi So, University of Southern California)**  
**Due:**

The last day of lecture at 11:59PM  
  
**Submission:**

The program file with comments, and readme documentation  
  
Email your project to:  
[email address removed]

With the subject line:  
ITP 125 – Programming Project

**Program Description:**

If you haven't already seen the Old Spice voice mail messages, watch them here:

Male version: http://www.youtube.com/watch?v=-8JsvwUcok0   
Female version: http://www.youtube.com/watch?v=Kx-78v6WLN8

Your assignment is to write a Python program that outputs an mp3 file voice mail message for the users own phone number. You will also allow the user to choose whether they want the female or male version, and which reasons they want to include in their message.  The spliced mp3 files will be provided for you. Please stick with the naming conventions used for the files, as this will make your grader's life much easier.

**Program Requirements:**

You must allow the user to input:

1. Whether they want the male or female version
2. Their 10-digit phone number. All of these forms should be acceptable:

* 012-345-6789
* (012) 345-6789
* 012.345.6789
* 0123456789

1. The user must be able to choose at least one reason from the list. The list of reasons they have to choose from must match the reasons listed for the male and female version. (If the user chooses male, you shouldn't display reasons in the female version of the voice mail). You have some freedom in how you choose to approach this, but here is one way to do it: output a list (1- 4, or however many reasons there are), and allow the user to choose the numbers of the corresponding reasons.   
     
   Example:

[1] Reason 1   
[2] Reason 2   
[3] Reason 3   
[4] Reason 4   
Enter the reasons you want in your voice mail (numbers 1-4): 134

1. The user must be able to choose at least one ending from the list. Similar to the reason, it must apply to whether they choose the male or female version. You may also want to implement the same menu-driven user input as is shown for the reason.
2. You must display a summary of the above information, and allow the user to either confirm or reject the settings shown. If the user rejects it, the program should restart to the beginning.   
     
   Hint:   
   If the program restarts, make sure to clear whatever files you listed to include in the final mp3.
3. If the user confirms, then you should ask what they want to call their output file name
4. You must also provide an app command line method by which to enter the above data. These  are the flags, which you will have to use:

-g (male/female)   
-n (phone number)   
-r (reasons)   
-e (endings)  
-o (output name)  
  
It is up to you if you want to require the user to list in this order, just make it clear to the user how it should work.

Example:

*myproject.py -g m -n 012.345.6789 -r 23 -e 13 -o voicemail.mp3*

1. You must output a file listing the file names of the sliced mp3 files that were used to make your final voice mail mp3 file. For example:  
     
   male 555-555-5555 f-beginning.mp3 f-op2.mp3 f-op3.mp3 f-op1.mp3 ... f-end.mp3
2. Finally, you must output an mp3 file with the file name specified by the user.
3. In order for the program to combine the MP3 files, it will need to download the files from the server:  
     
   <http://www-bcf.usc.edu/~chiso/itp125/project_version_1/>  
     
   the program cannot pre-download all the MP3. It will only download the needed MP3s after the user makes all the selections. After combining the MP3s, the program will remove downloaded files.
4. For the program to work, it will need to detect if the program is being run on Windows or OSX. The commands to combine MP3s for Windows/OSX are different.

**Input Validation**

You must validate the user's input. For example, if the user must type 1 for male, or 2 for female, and if they enter 3, or “asdf,” then you must inform them that they've entered an invalid input, and then restart the program. This will be a bit more challenging in the reasons and ending sections; make sure to check for if the user enters something like 1458.

**Documentation**

Your code must be commented. You should describe what your functions do, what your variables represent, or anything else that isn't immediately obvious. In addition to commenting your code, you must provide a readme file that describes the logic of your program. The readme file has to be in txt or pdf format.

**Tips and Help**

* Combining MP3s in Windows and Linux (this will work in OSX as well)  
  <http://stream-recorder.com/forum/join-multiple-mp3-files-t4656.html>
* Downloading files over the Internet with Python  
  <http://stackoverflow.com/questions/22676/how-do-i-download-a-file-over-http-using-python>
* Handling arguments from the command line in Python  
  <http://docs.python.org/library/getopt.html>
* Detecting the OS from Python  
  <http://stackoverflow.com/questions/1854/how-can-i-tell-what-os-i-am-running-on-from-python>
* Handling Exceptions and Writing Files in Python  
  <http://www.diveintopython.net/file_handling/index.html>
* Executing a command in the OS using Python  
  <http://stackoverflow.com/questions/89228/how-to-call-external-command-in-python>

**Grading Breakdown Programming** **(50 points):**

5 points – Being able to download a MP3 across the network

20 points – Being able to combine the downloaded MP3s into the proper voicemail ask by the user

5 points – Making sure all temporary files that were created are removed. In other words, the only file that should be left after running the program is your program, the voice message MP3, and the output txt file.

10 points – Command line arguments work with the program and the user is not forced to use the menu system.

10 points – Was an output file with the requested MP3s generated.

20 points – Error Checking this will be tested with invalid inputs on the command line arguments. If the command line arguments fail, the menu system will be tested.

30 points – Documentation:

15 points – comments in the code

15 points – readme file

Questions? Concerns? Email the TA or grader