

This is **not** a collaborative assignment; you must design, implement and test the solution(s) on your own. You may not consult or work with anyone other than the course instructor or TAs. In addition, you may not include solutions or portions of solutions obtained from any source other than those provided in class. Obtaining or *providing* solutions to any homework problems for this class is considered academic misconduct. If you are not sure what this means, consult the class syllabus or discuss it with the course instructor.

This assignment requires writing a single Python script that must be submitted online *prior* to the due date/time. Late submissions will not be accepted. Name your source code: `hw8.py` Submit your source code file using the appropriate homework submission link on the Moodle website.

The total point value for programming assignments will be awarded for solutions that are *complete, correct, and well constructed*. A "well constructed" program entails good design, appropriate comments and general readability (descriptive names for variables and procedures, appropriate use of blank space, etc.). The following will result in a score reduction equal to a percentage of the total possible points:

- Incorrectly named/submitted source file (10%)
- Constraints not followed (40%)
- Failure to execute due to syntax errors (30%)

Note that your work will be graded using, and must function correctly with, the current version of Python 3 on CSE Labs UNIX machines. If you complete your programming assignment using a different system, it is your responsibility to ensure your programs work on CSELabs machines *prior* to submitting them.

A. (40 points) **Flight Data Filter**

A *filter* program is one that reads each record of an input file and creates a new output file containing only those records that match some qualifying criteria. For this assignment, you will construct a short Python *filter* program that will process real-world airline flight-routing data and produce a file containing only those records that represent flights that arrive and/or depart from a particular airport.

Use your web browser to view the following URL:

<http://openflights.org/data.html>

This is a website that maintains information on airline flights. Browse the site, locate the section entitled: "Route Database" and download the `routes.dat` file (~2MB). This file describes all worldwide airline flight segments between various airports. You should also review the text that describes the content and format of the file.

Write a Python program that will process this file and produce a smaller file containing only those flights that originate in, or depart from, a user-specified airport. Your program should solicit the name of the input file, the name of the (filtered) output file and the IATA letter code for the selected airport. For example, Minneapolis is MSP and San Francisco is SFO, etc.

Requirements:

Your program must do the following:

- Solicit the name of the input route-data file from the user.
- If the file does not exist or fails to open, continue to solicit the file name a maximum of 3 times and then terminate the program
- Solicit the name of the output (filtered) route-data file from the user.

- If the file exists, then ask the user if he/she would like to overwrite the existing file. If not, then terminate the program.
- Solicit an IATA airport code from the user (e.g., MSP, SFO, etc.). Your program should accommodate entry in either upper or lower case.
- Process the input file and produce an output file consisting only of the records of those flights that originate or depart from the indicated airport.

Constraints:

- Do not import/use the Python `csv` or `shutil` modules. All other standard Python modules are acceptable, including `os`
- You must use the `.readline()` method to process the input file records. Do not use a `for` loop to process file data for this assignment.
- All files must be properly opened and closed.

Example:

```
Airport Routing Filter
Enter the source file name: routes.dat
Enter the output file name: mspdata
Enter airport symbol: msp
Finished
>>> ===== RESTART
>>>
Airport routing filter
Enter the source file name: mickeymouse
File not found. Reenter: routes.dat
Enter the output file name: mspdata
File exists... overwrite? (y/n): y
Enter airport symbol: msp
Finished
```