ME 498 K; Spring 2023 — Homework Reflections and Revisions

Aidan Hunt (ahunt94@uw.edu) — Due: June 9, 2023

In this final assignment, you will revisit the programs you have written over the course of the quarter. Submit your answers to the Homework Reflection questions, as well as descriptions of your homework revisions, as a single PDF document. Submit your revised homework scripts as individual `.py` or `.ipynb` files to GradeScope.

1. Homework Reflection (Required)

The following questions describe hypothetical scenarios in which we would want to expand upon the code you have already written. In **3-5 sentences**, answer each question by describing, in general terms, the type of code you would write to address the scenario. For example, you could describe the new functions you might define, or parameters and/or returns you might add to your existing functions, or the potential challenges you might run into. You do not need to write any code, nor am I looking for a "right" answer or a fully-functional solution; as long as you engage with the question and describe your ideas, my grading will be lenient.

Homework 1. Describe a hypothetical upgrade to your creative programming script that uses something new that you learned in this course. This upgrade could add new functionality to your code, or could perform an existing task in a more efficient/elegant way.

Homework 2. Imagine that we would like to upgrade our text parser to recognize words with common roots. For example, "engineer", "engineers", and "engineer's" would all be sorted and counted as "engineer". Describe the logic you might add to your text parsing code to account for this.

Homework 3. Imagine that we would like to add a fifth channel type, the parabola (equations in HW3 spec) to the open channel flow assessment. What new functions would you define to add this functionality? What existing functions could you repurpose?

Homework 4. Imagine that we would like to update `computeAirfoils()` so that it can compute the coordinates for both symmetric NACA airfoils and cambered NACA airfoils. What kinds of modifications would you need to make to your code to incorporate this new set of equations? What parts of your code could remain unchanged?

Homework 5. Identify a Python package that we did not discuss in this class (some examples include SciPy, SymPy, Scikit-Learn, Geopandas, etc...). What special functionality does this package bring to Python? Describe an engineering problem you might use this package to solve.

Homework 6. Consider that we want to process the data in the files from other sensors (for example, the encoder data in row 14 of each modelData file), in addition to that of the two load cells. How might you approach this while making few adjustments to your batch-processing loops? What additional classes or functions might you define?

Homework 7. Describe a figure you could make related to your research, capstone project, or extracurricular interests that could leverage color to communicate a key result or observation (e.g., using color to show trends in free-surface deformation like in Lectures 16/17). What plotting function(s) would you use? What type of colormap would be well suited to the data? **Extra credit (10 points):** create the figure and upload your code and any associated data to GradeScope.

2. Homework Revision (Optional)

Being a good programmer doesn't mean that you always write perfect code on the first try – often we steadily refine our scripts over time as we learn new techniques or new needs arise. To practice this, **you may revise your script(s) from any of your homework submissions to earn points back on any deducted categories**. A deduction is something marked "-1", "-2", etc. in your Gradescope feedback ("-0" is not a deduction).

For **each** deduction within an assignment that you would like to earn points back on, do the following:

1. Implement my feedback in your code. (If you are not sure what my feedback is suggesting, ask me!)
2. In **1-2 sentences**, describe the specific change(s) you made to your code and how it addresses the output error or style that the point was originally deducted for. So that I know what feedback you are responding to, include the text of the corresponding GradeScope comment in your submission (see example on next page).

Some additional guidelines:

- You can choose to address any number of deductions in your revisions for each assignment. For example, if you lost 5 points on HW2, you could choose to focus your revisions on a subset of the deductions such that you earn 3 of those 5 points back.
- Similarly, you do not have to submit revisions for all assignments. For example, if you lost points on both HW1 and HW2, you could submit revisions for both HW1 and HW2, only HW1, or only HW2.
- You can only gain points by revising your scripts — you cannot lose points that you had previously earned.
- To receive credit for a revised script, you must have submitted the corresponding assignment when it was originally due (late days okay).
- To receive credit for your revisions, your script must still run without error after you incorporate your changes. If your revisions in one section of your code require another section to be adjusted in order to avoid errors, you should do so.

Example of reflection and revision responses for Homework 2

Homework 2: Reflection:

For my text parser to be able to detect root words, when processing the text file, I would add a check for... I would define a function that... I would add a parameter to `exampleFunctionName()` that represents...

Homework 2: Revisions

-1: The way you are filtering punctuation and digits isn't exactly correct. You are replacing punctuation with spaces, which means that hyphenated words will be counted as two words, rather than as one word as requested by the spec.

I have adjusted updated the punctuation replacement routine to replace punctuation with empty strings. This allows for hypthenated words to be counted as single words.

-1: Something else important to mention in your docstring and program summary is that punctuation/digits are not considered when counting/sorting words.

I have added language to my file-reading function's docstring that describes that punctuation is removed from the words when the file is read.