

Charles V Fisher
ITIN8000 FA2021
HW2 Reflection

Github Repo Link:

<https://github.com/cfisherCPL/ITIIN-8000-Assignments-Charles-Fisher>

Write a 2-3 short paragraph reflection on your experience while working on this homework. Things to consider including are:

1. What seems to be the pros and cons of CSVs vs JSONs?
2. Did you parallelize your code? How would parallelizing it affect Part 2?
3. What things helped you work well?
4. Are there things you would have done differently if you had to redo the assignment from scratch to make your life easier?

CSVs tend to be easily legible and portable to a number of gui apps for simple inspection compared to JSON. With Pandas installed both are wicked easy to work with, and that ended up being a big chunk of my brain fog on working with all of this. The basic CSV and JSON packages work similarly to java in keeping the document stream open or closed and that made general sense and made some very pretty looking JSON. Took more lines of code than Pandas, but made sense to me. Pandas just sorta...does it all with a function call, but the json it put out was ugly to parse out visually. Still, it just worked when testing by re-writing it to a csv with pandas.

I did not parallelize my code simply due the attempt to get the assignment in not-as-late. However, the flowchart and architecture diagram showcase HOW this could be done. Given that the dataset csvs to be fed in are pretty darn large, reading them each into their own panda object at the same time could have marginally sped up work overall. Pushing the read-in and output selected columns function to asynch using multiprocessing library woulda been pretty darn cool, and most likely necessary for larger datasets.

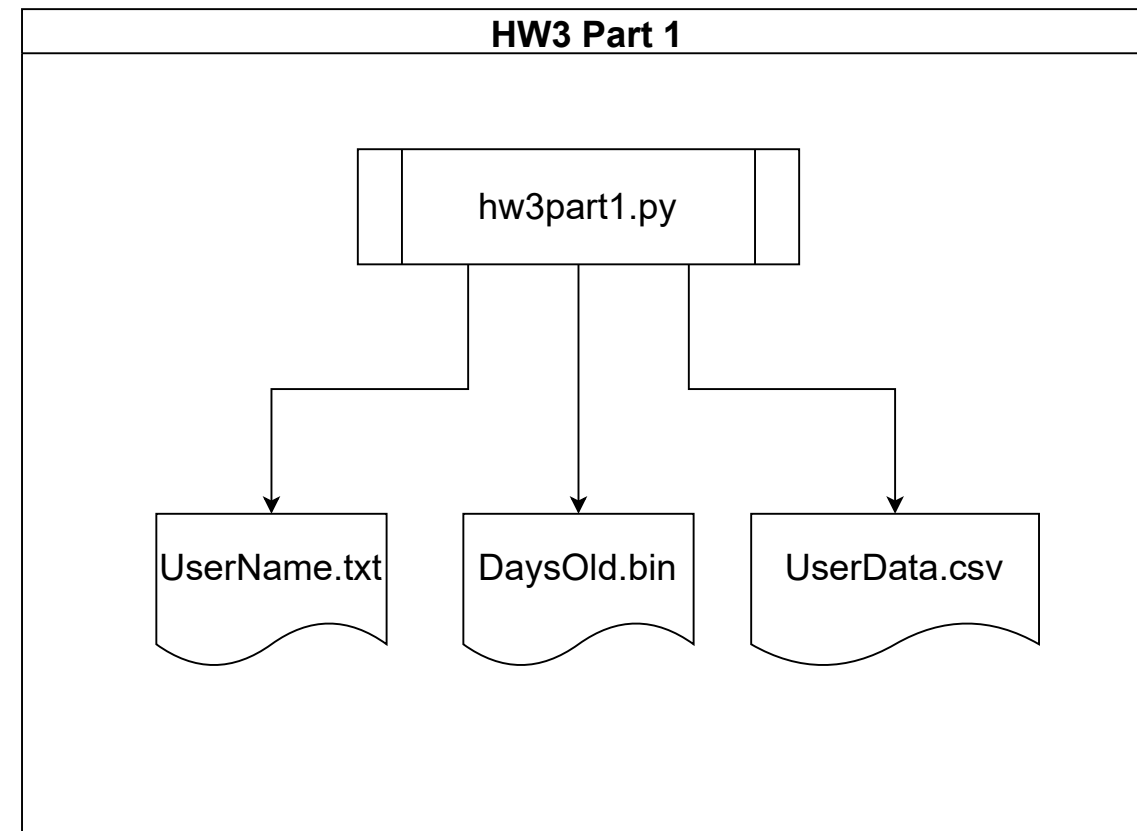
Pandas. Without a doubt this lil library saved so much hubbub that I'm actually upset I spent so much time trying to NOT use it and do the work exclusively through the csv and json imports.

I should have simply trusted the json output coming from the pandas write function. Even though it looked un-pretty, it is MORE than fully functional and portable for use. Really that just goes to show how parsing large sets of data really isn't a thing for human eyes! I would have liked to get to practice parallelizing this one ahead of our work with MachineLearning. The early version of part 2 didn't take advantage of pandas at all and I wasted days needlessly trying to brute through what should have been a file

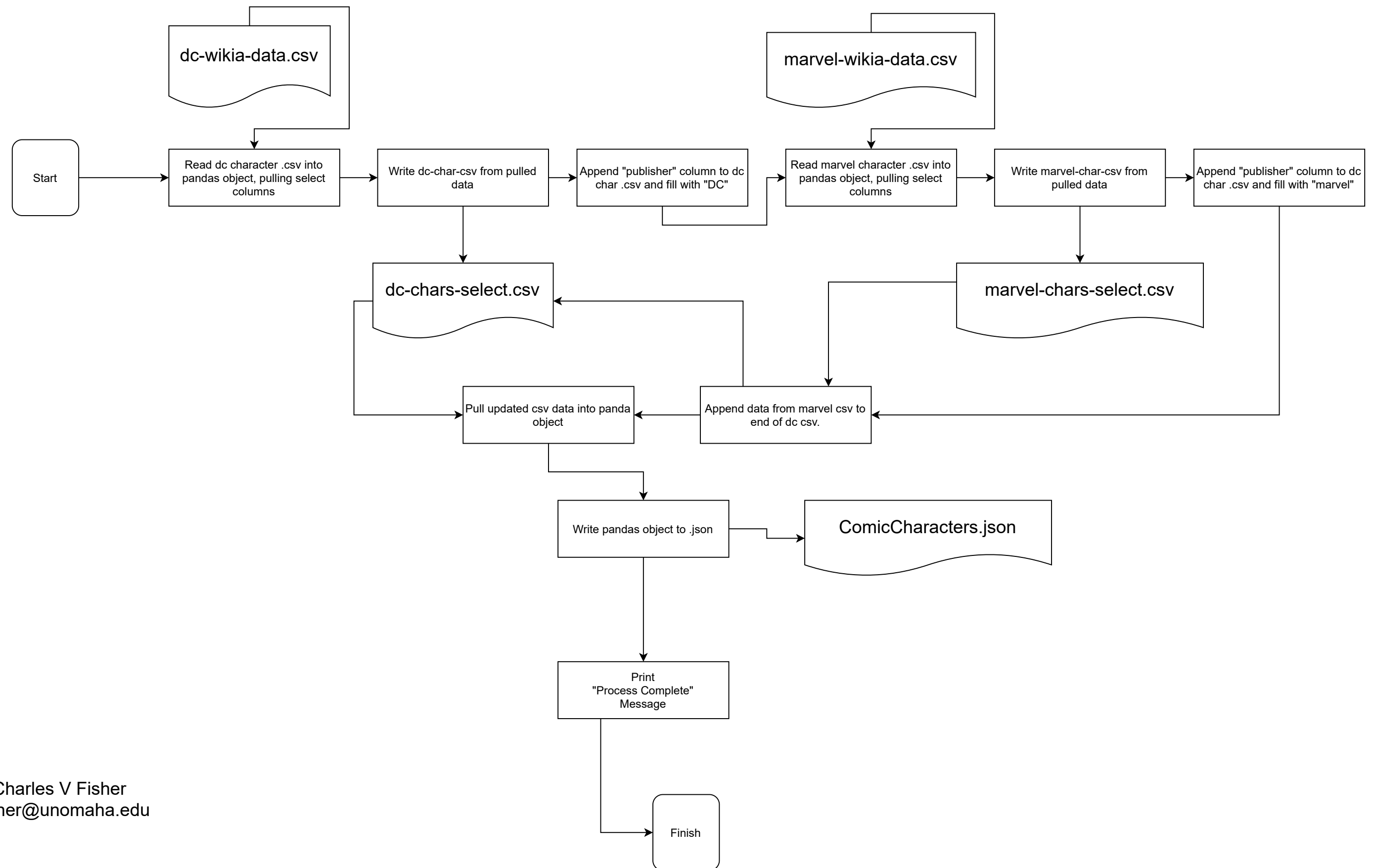
Homework 3 Part 1
Flowchart



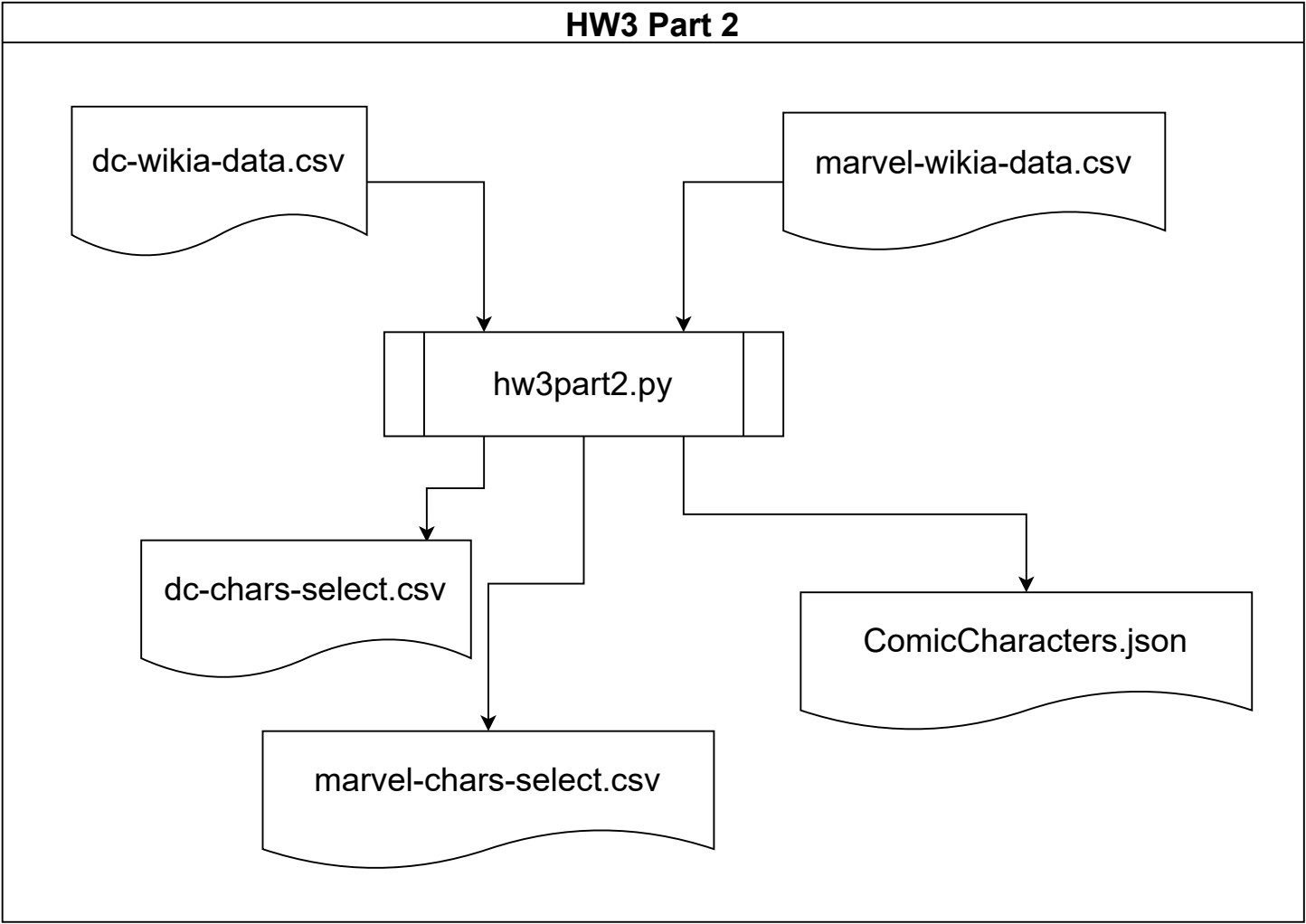
Homework 3 Part 1
Architecture Diagram



Homework 3 Part 2 Flowchart



Homework 3 Part 2
Architecture Diagram



```
1 """
2 Description from canvas:
3 Imports the data from dc-wikia-data.csv and marvel-wikia-
  data.csv
4 Combines them into a single JSON file called ComicCharacters
  .JSON
5 The JSON file should list every character as an object by "
  Character Name"
6 Must Contain Object Ownership containing the value Publisher
  (DC or Marvel)
7 Must contain the object Characteristics containing the
  values:
8 Alignment (Good, Bad, or Neutral)
9 Eye Color
10 Hair Color
11 Gender
12 When run it should generate a new file
13 """
14 # import pandas
15
16 import pandas as pd
17
18
19 # use pandas to read the DC csv and pull the columns we need
20 # write a new csv from pandas object
21
22
23 # add ending column to ID entries as DC
24
25 # use pandas to read the Marvel csv and pull the columns we
  need
26 # write a new csv from pandas object
27
28 # add ending column to ID entries as Marvel
29
30 # append marvel csv to the end of dc csv
31
32 # write merged csv into ComicCharacters.json
33
34
```

```
35 # for concurrency, let's make DC and Marvel happen  
    separately, then append them?  
36 # stretch goal we've already missed. damnit.  
37  
38  
39
```

```
1  """
2  HW pt1 Description Provided on Canvas
3  Ask for the users first and last name, favorite color, and
   date of birth
4  Calculate how many days old the user is
5  Write the user's name in the format last name, first name
   and favorite color in a .txt named UserName
6  Write the user's age in days to a binary file named Days Old
7  Add the user to a CSV file named UserData.csv in the order
   Last Name, First Name, Favorite Color, Days Old
8  A file named hw3part1.py should perform these tasks when it
   is run
9  """
10
11 # Import datetime (from datetime import date)
12 # Import csv
13
14
15
16 # prompt first name
17
18 # prompt last name
19
20 # prompt favorite color
21
22 # prompt DoB
23 # make sure to prompt with format note!
24
25 # Calculate how many days old the user is and stash as
   variable
26
27 # do the file work below here:
28 # create .txt file UserName with open()
29
30 # Write the user's name in the format last name, first name
31 # and favorite color in a .txt named UserName
32
33 # close .txt file
34
35 # create and open filename.bin in wb mode
```



```
36
37 # Write the user's age in days to a binary file named Days
    Old
38
39 # close .bin file
40
41 # create list from user input as shown:
42 # Last Name, First Name, Favorite Color, Days Old
43
44 # create and open filename.csv
45
46 # create the csv writer object
47
48 # use writer to write row from list
49
50 # close csv
51
52
```