Sam Ormerod
2020-Aug-26
Foundations of Programming: Python
Assignment 07

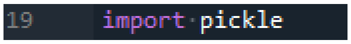
CD Inventory Round 4

Intro

We looked into new ways of reading and writing data in text and binary format. We also touched on structured error handling which was really straight forward given the built in error classes in Pythons. Lab07_B revisited adding functionality to select the operation mode via program arguments. Ultimately Lab07_B took up the majority of my time and focus this week. I struggled to comprehend exactly what was being asked of us. We received confirmation of what was being asked during class and I was able to complete the lab shortly after. The assignment this week wasn't very intensive but still had its challenges.

Pickles

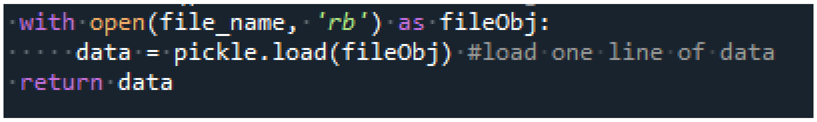
Reading and writing to binary files[1] known in Python as pickling[2] is fast and easy. When working with binary files we don't have to format the data when writing to the file or when reading the file into memory. In order to utilize this functionality we first need to import the pickle module please see Figure 1.



```
import pickle
```

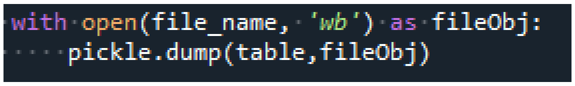
Figure 1 - Importing the pickle module

We must indicate we are going to read and write binary by including a 'b' in our calls, please see Figure 2 and Figure 3. Respectfully this is signified by 'rb' and 'wb'.



```
 with open(file_name, 'rb') as fileObj:
     data = pickle.load(fileObj) #load one line of data
 return data
```

Figure 2 - Reading pickled binary data into memory



```
with open(file_name, 'wb') as fileObj:
    pickle.dump(table,fileObj)
```

Figure 3 - Writing pickled data to a file

[1] FDN_Py_Module_07.pdf page 14
[2] https://docs.python.org/3/library/pickle.html

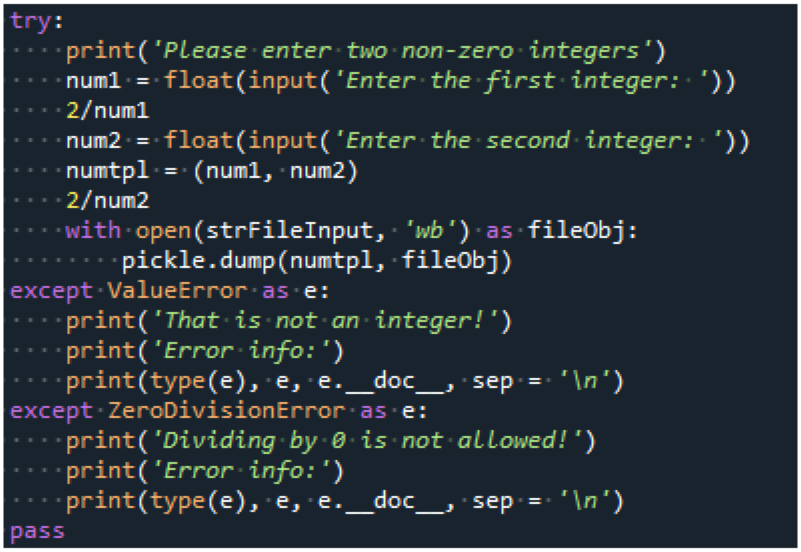
I found this link to be useful to gather further information regarding the pickle module.

https://docs.python.org/3/library/pickle.html

Structured Error Handling

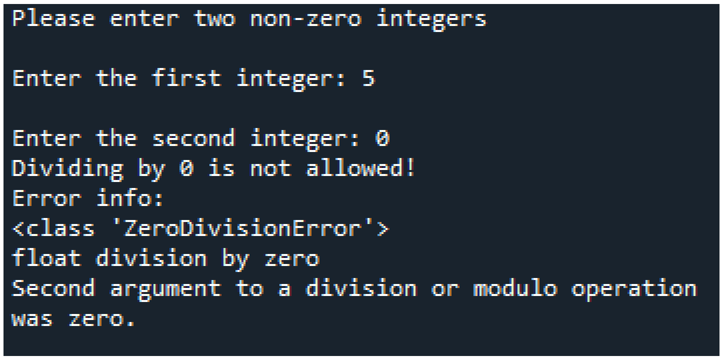
Utilizing the built in structured error handling classes in Python we are able to provide the user with instant feedback regarding issues that arise. I found the following links to be useful when looking into the built in exception regarding error handling that Python has. Please see Figure 4 and Figure 5.

https://docs.python.org/3/library/exceptions.html
https://docs.python.org/3/tutorial/errors.html



```python
try:
    print('Please enter two non-zero integers')
    num1 = float(input('Enter the first integer: '))
    2/num1
    num2 = float(input('Enter the second integer: '))
    numtpl = (num1, num2)
    2/num2
    with open(strFileInput, 'wb') as fileObj:
        pickle.dump(numtpl, fileObj)
except ValueError as e:
    print('That is not an integer!')
    print('Error info:')
    print(type(e), e, e.__doc__, sep = '\n')
except ZeroDivisionError as e:
    print('Dividing by 0 is not allowed!')
    print('Error info:')
    print(type(e), e, e.__doc__, sep = '\n')
pass
```

Figure 4 - Examples of the built in error handling



```
Please enter two non-zero integers

Enter the first integer: 5

Enter the second integer: 0
Dividing by 0 is not allowed!
Error info:
<class 'ZeroDivisionError'>
float division by zero
Second argument to a division or modulo operation
was zero.
```

Figure 5 - Example of figure 4 processing an error

The assignment was a continuation of our CD inventory incorporating pickling and error handling. Please see the Appendix for screenshots of my code.

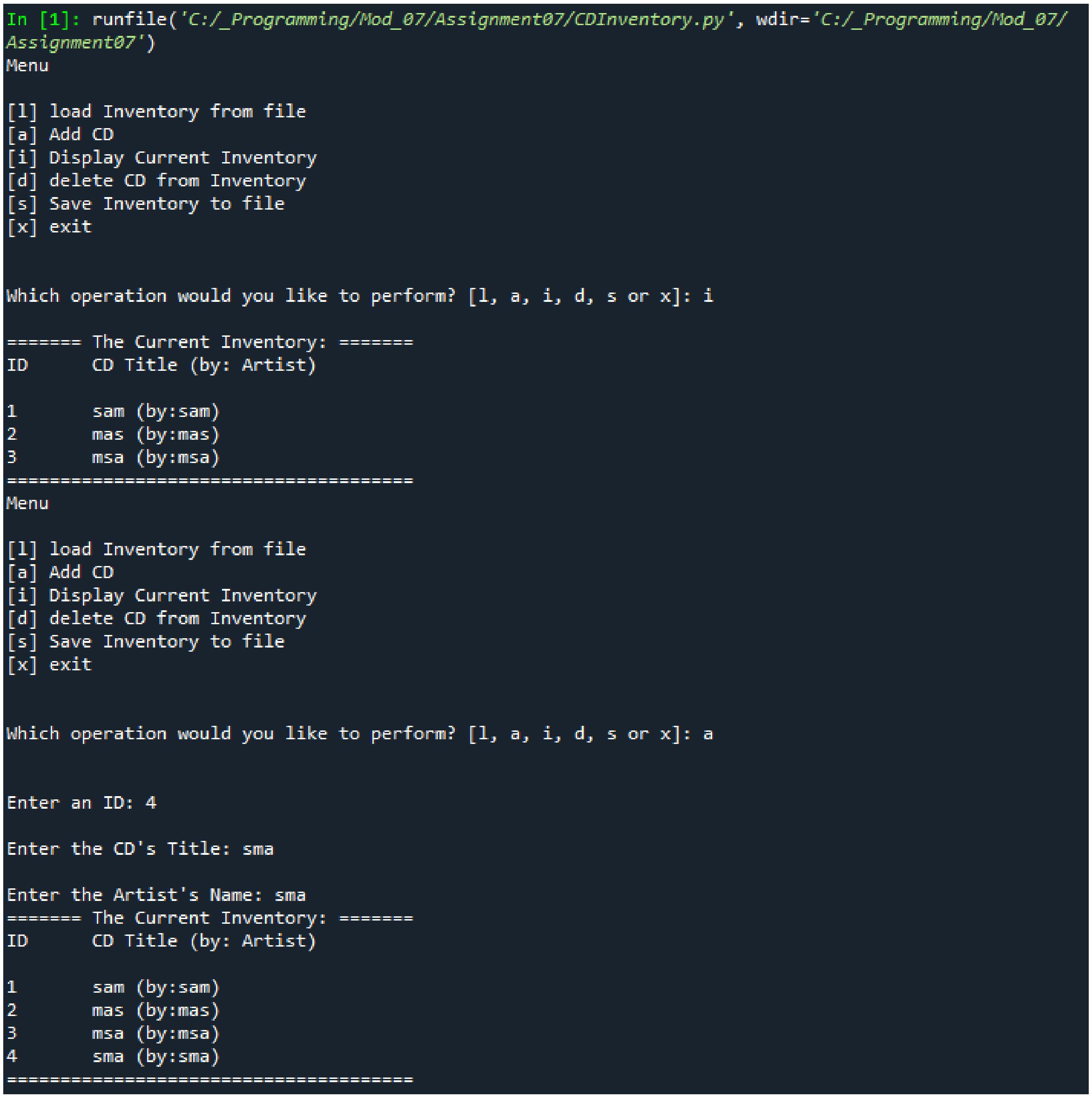
Summary

The assignment and new concepts were really straight forward this week. While I struggled to understand what was being asked to complete the Labs. Once I received clarification on what exactly being asked it did not take me long to successfully code the Labs. This course has progressed really fast. I believed that I have kept a firm grasp on everything we've gone over but this week has exposed areas that I should revisit.

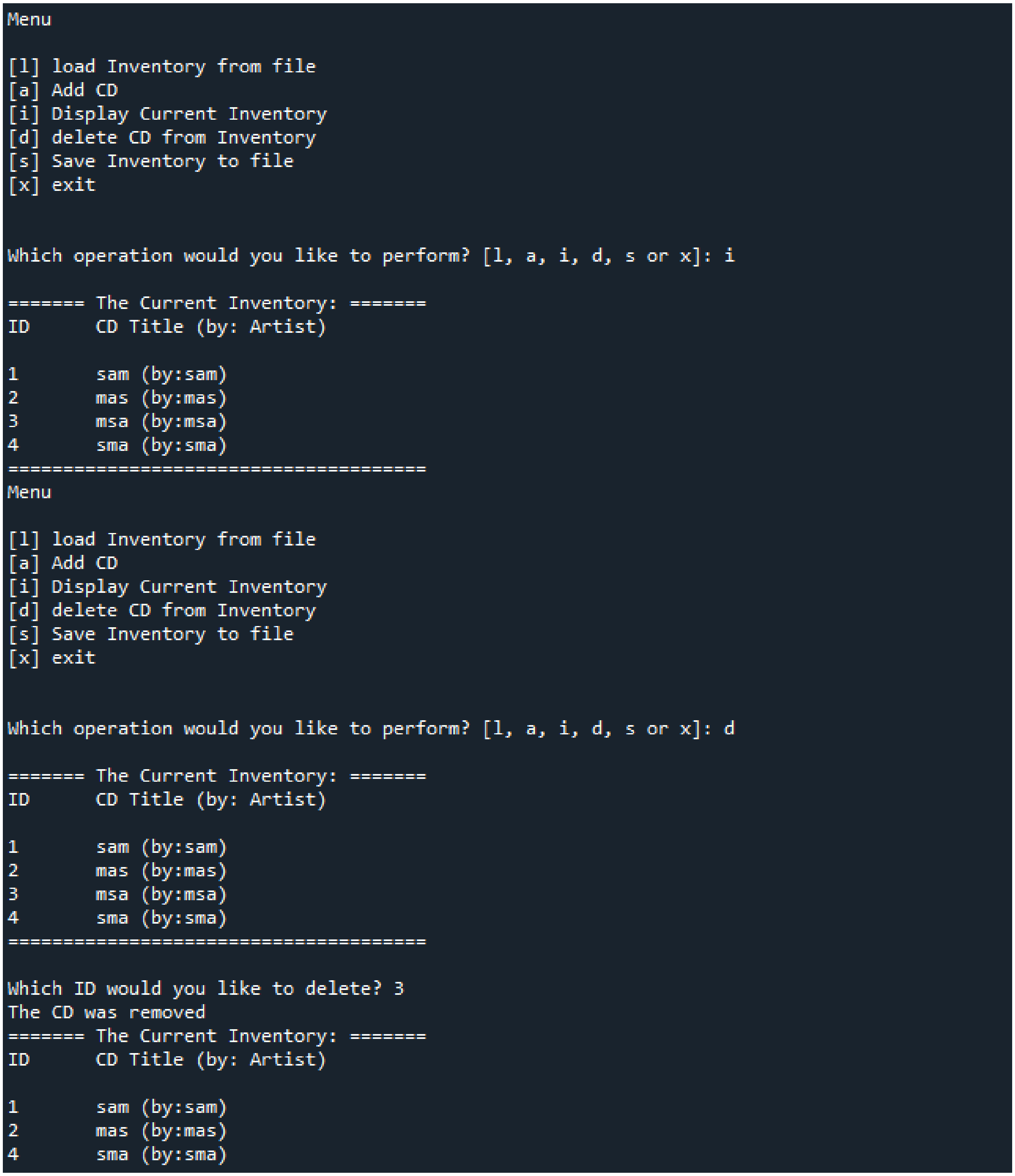
Appendix

Full program available on GitHub

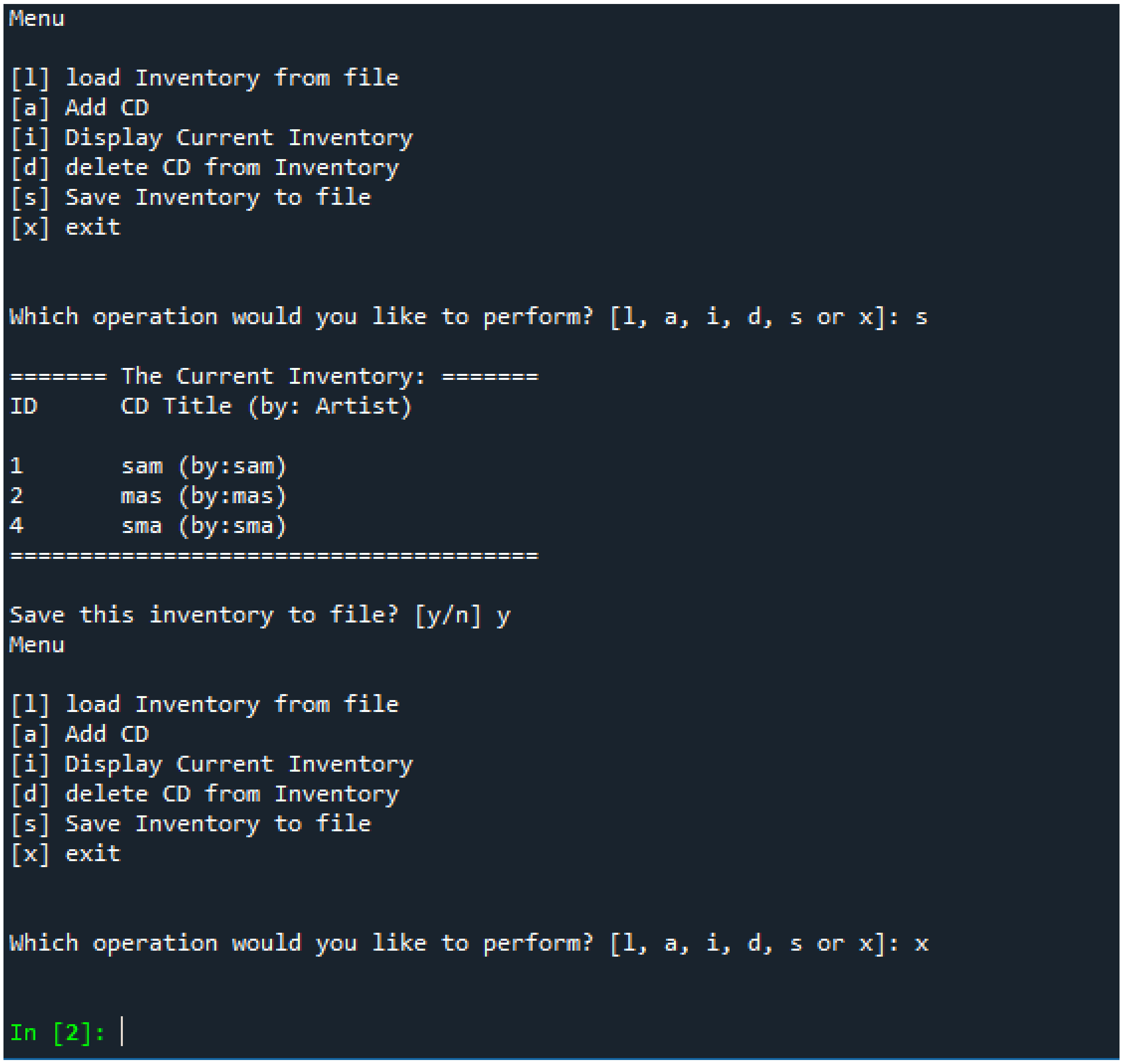
Spyder Screenshots:



```
In [1]: runfile('C:/_Programming/Mod_07/Assignment07/CDInventory.py', wdir='C:/_Programming/Mod_07/
Assignment07')
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit


Which operation would you like to perform? [l, a, i, d, s or x]: i

======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       sam (by:sam)
2       mas (by:mas)
3       msa (by:msa)
======================================
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit


Which operation would you like to perform? [l, a, i, d, s or x]: a


Enter an ID: 4

Enter the CD's Title: sma

Enter the Artist's Name: sma
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       sam (by:sam)
2       mas (by:mas)
3       msa (by:msa)
4       sma (by:sma)
======================================
```



```
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit


Which operation would you like to perform? [l, a, i, d, s or x]: i

======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       sam (by:sam)
2       mas (by:mas)
3       msa (by:msa)
4       sma (by:sma)
======================================
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit


Which operation would you like to perform? [l, a, i, d, s or x]: d

======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       sam (by:sam)
2       mas (by:mas)
3       msa (by:msa)
4       sma (by:sma)
======================================

Which ID would you like to delete? 3
The CD was removed
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       sam (by:sam)
2       mas (by:mas)
4       sma (by:sma)
```



```
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit


Which operation would you like to perform? [l, a, i, d, s or x]: s

======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       sam (by:sam)
2       mas (by:mas)
4       sma (by:sma)
======================================

Save this inventory to file? [y/n] y
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit


Which operation would you like to perform? [l, a, i, d, s or x]: x


In [2]: 
```