Ljubica Trpcheva Smileski

November 25, 2022

Foundations of Programming, Python

Assignment 07

# CD Inventory program with functions, Error Handling and Pickling feature

## Introduction

This Module 07 we got an introduction to Error Handling in Python and Pickling feature - working with binary files. We found that illegal operations can raise exceptions. There are plenty of built-in exceptions in Python that are raised when corresponding errors occur. In those situations, Error Handling is very handy and can be used to notify the user why the error occurred and exit the process that caused the error.

## Research Exception Handling in Python

After researching Exception handling in Python, there are websites that helped me understand that exceptions can be handled using a try statement, that the critical operation which can raise an exception is placed inside the try clause and the code that handles the exceptions is written in the except clause. They are listed below:

https://www.programiz.com/python-programming/exceptions

https://realpython.com/python-exceptions/

https://python-course.eu/python-tutorial/errors-and-exception-handling.php

https://www.geeksforgeeks.org/python-exception-handling/

this tutorial was not very helpful:

https://www.tutorialspoint.com/python/python_exceptions.htm

this tutorial was not easy to follow:

https://desktop.arcgis.com/en/arcmap/latest/analyze/python/error-handling-with-python.htm

## Research Pickling in Python

After researching Pickling in Python, I found websites that helped me understand that any object in Python can be pickled so it can be saved on disk. Pickling is converting a Python object (list, dictionary, etc.) to a character stream. They are listed below:

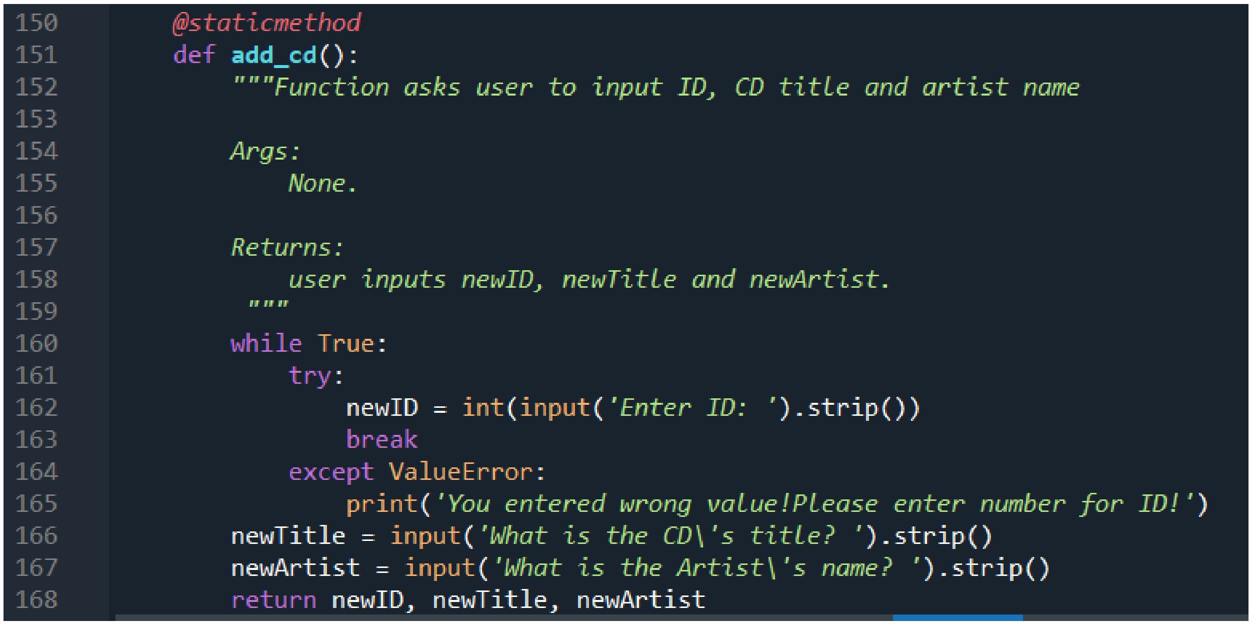
https://www.geeksforgeeks.org/understanding-python-pickling-example/

https://www.digitalocean.com/community/tutorials/python-pickle-example https://realpython.com/python-pickle-module/

## CD Inventory program updated with error handling

This week for Assignment 7, we are using our CDInventory.py from last week's assignment. I started with testing and inputting wrong data, to see where the errors occurred. Next, I started adding Error handling code to fix those errors.

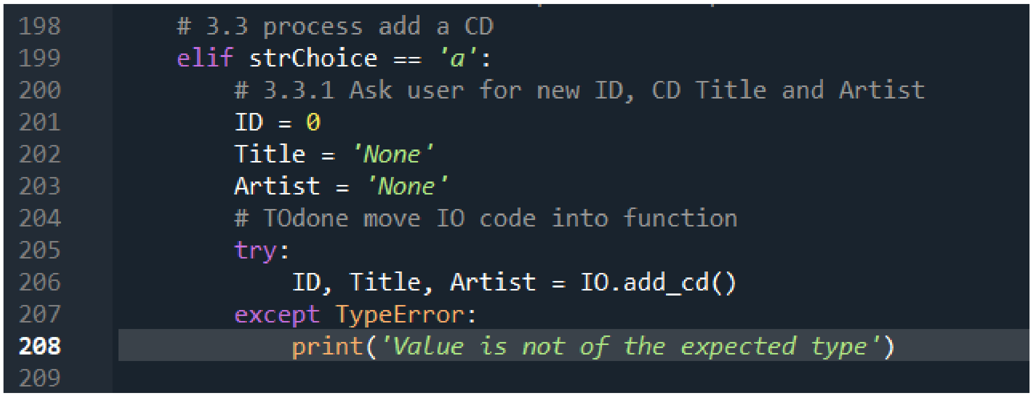
The fists error occurred in the function add_cd(), where I input string instead of an integer for newID. I used 'try: except ValueError: 'statement to handle this error. ValueError is raised when a function receives an argument of the correct type but an inappropriate value. See Figure 1.



```
    @staticmethod
    def add_cd():
        """Function asks user to input ID, CD title and artist name

        Args:
            None.

        Returns:
            user inputs newID, newTitle and newArtist.
         """
        while True:
            try:
                newID = int(input('Enter ID: ').strip())
                break
            except ValueError:
                print('You entered wrong value!Please enter number for ID!')
        newTitle = input('What is the CD\'s title? ').strip()
        newArtist = input('What is the Artist\'s name? ').strip()
        return newID, newTitle, newArtist
```

*Figure 1 – Structured Error Handling (ValueError)*

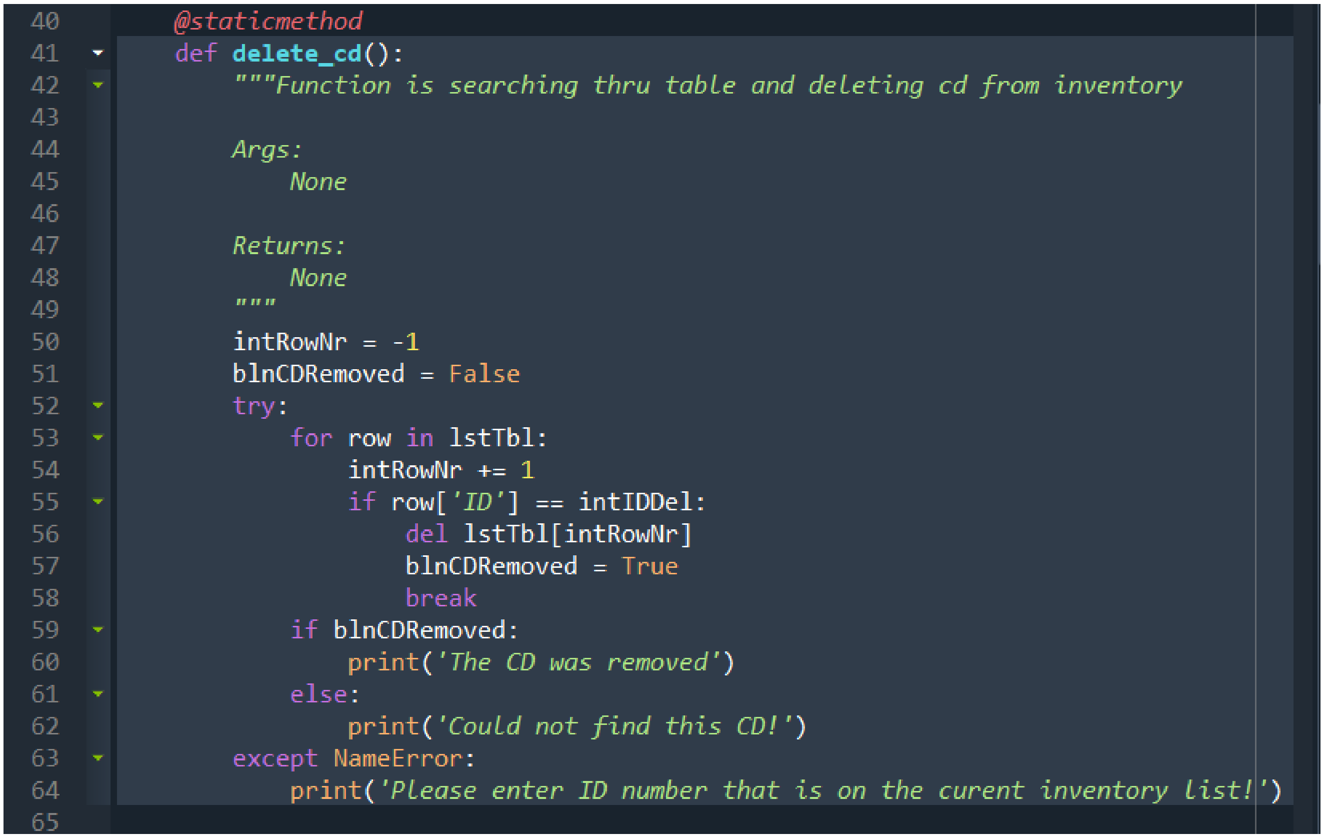
The second error occurred in the main menu when I call the function add_cd(), because of inputting string instead of an integer for ID. I used 'try: except TypeError:' statement to handle this error. The TypeError object represents an error when an operation could not be performed, typically when a value is not of the expected type. See Figure 2.



```
    # 3.3 process add a CD
    elif strChoice == 'a':
        # 3.3.1 Ask user for new ID, CD Title and Artist
        ID = 0
        Title = 'None'
        Artist = 'None'
        # TOdone move IO code into function
        try:
            ID, Title, Artist = IO.add_cd()
        except TypeError:
            print('Value is not of the expected type')

```

*Figure 2 – Structured Error Handling (TypeError)*

The next error occurred in the function delete_cd(), where I tried deleting CD that does not exist in my CD Inventory file. I used 'try: except NameError:' statement to handle this error. The NameError occurs when user tries to use a variable, function, or module that doesn't exist or wasn't used in a valid way. See Figure 3.

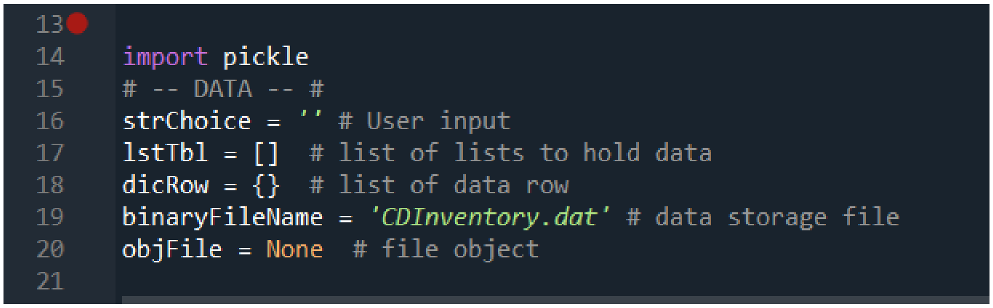


```
    @staticmethod
    def delete_cd():
        """Function is searching thru table and deleting cd from inventory

        Args:
            None

        Returns:
            None
        """
        intRowNr = -1
        blnCDRemoved = False
        try:
            for row in lstTbl:
                intRowNr += 1
                if row['ID'] == intIDDel:
                    del lstTbl[intRowNr]
                    blnCDRemoved = True
                    break
            if blnCDRemoved:
                print('The CD was removed')
            else:
                print('Could not find this CD!')
        except NameError:
            print('Please enter ID number that is on the curent inventory list!')
```

*Figure 3 – Structured Error Handling (NameError)*

## Pickling - Using Binary files

I modified the permanent data store to use binary data. First, I added 'import pickle' module to be able to use binary files. I modified my previously .txt file into .dat file (binaryFileName = 'CDInventory.dat'). See Figure 4.



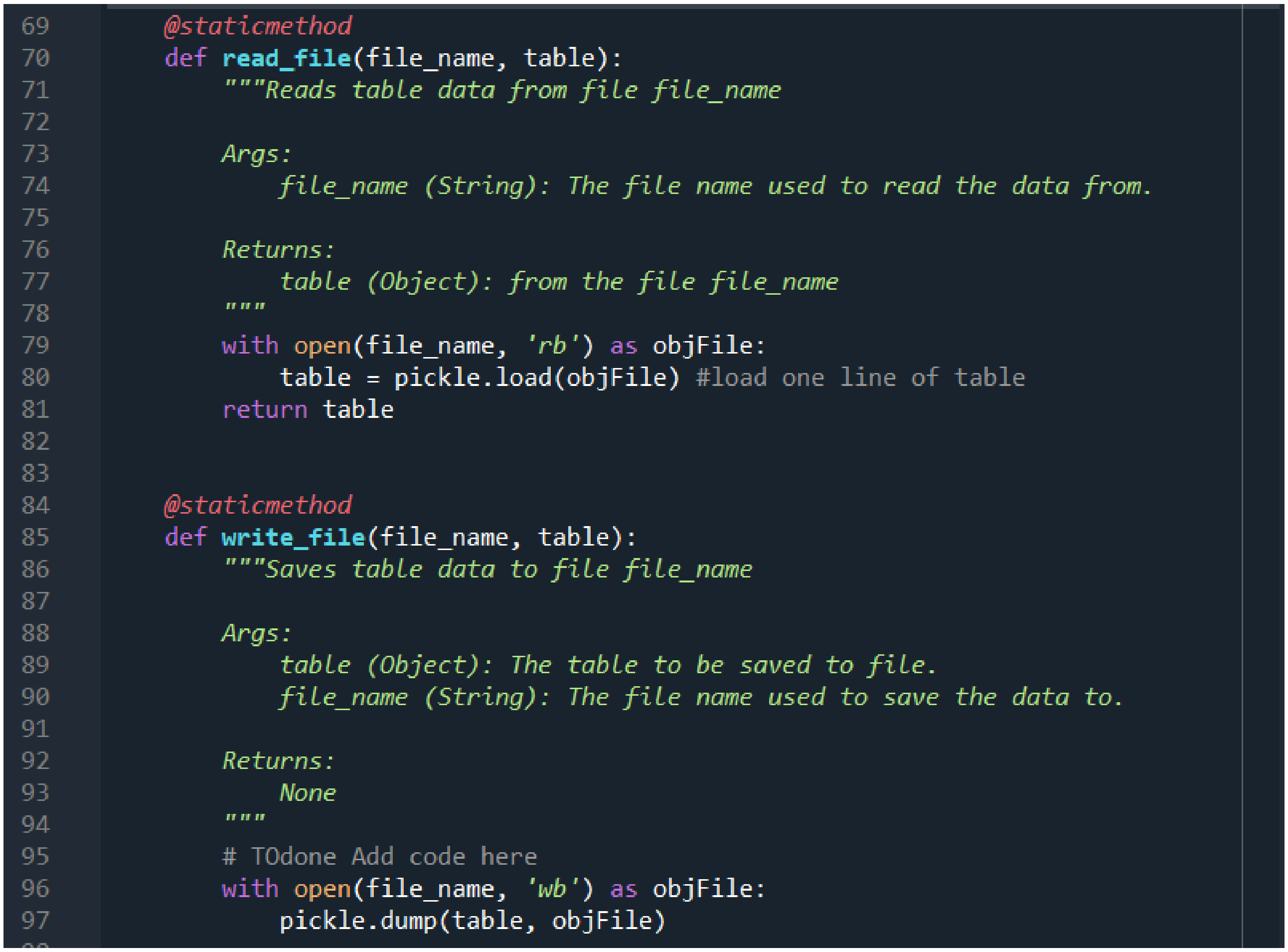
```

import pickle
# -- DATA -- #
strChoice = '' # User input
lstTbl = []  # list of lists to hold data
dicRow = {}  # list of data row
binaryFileName = 'CDInventory.dat' # data storage file
objFile = None  # file object

```

*Figure 4 – Pickling*

Then, I modified my write_file() function, using "wb" mode. "wb" mode opens the file in binary format for writing. Then used pickle.dump(), for storing object data in the binary file.

Next, I modified my read_file() function, using "rb" mode. "rb" mode opens the file in binary format for reading. When the file is open for reading, use pickle.load() to load pickled data from a file like object.

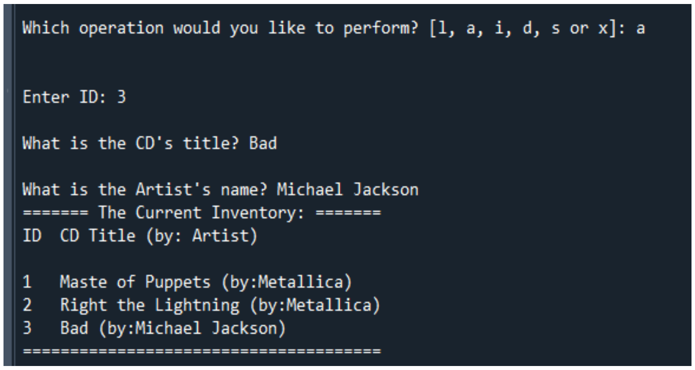
Unlike text files, binary files are not human-readable. See Figure 5



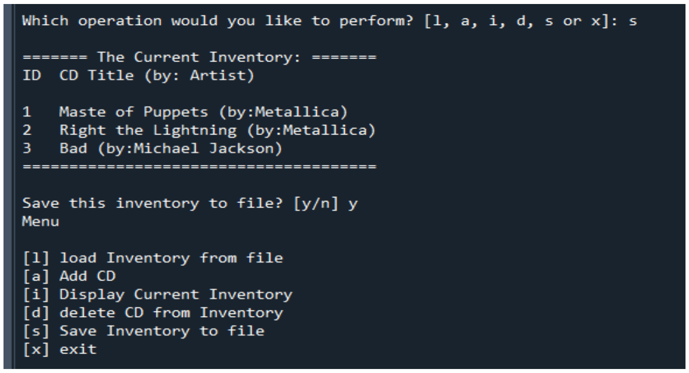
```
    @staticmethod
    def read_file(file_name, table):
        """Reads table data from file file_name

        Args:
            file_name (String): The file name used to read the data from.

        Returns:
            table (Object): from the file file_name
        """
        with open(file_name, 'rb') as objFile:
            table = pickle.load(objFile) #load one line of table
        return table


    @staticmethod
    def write_file(file_name, table):
        """Saves table data to file file_name

        Args:
            table (Object): The table to be saved to file.
            file_name (String): The file name used to save the data to.

        Returns:
            None
        """
        # TOdone Add code here
        with open(file_name, 'wb') as objFile:
            pickle.dump(table, objFile)
```

*Figure 5 – Read from and write to binary file*

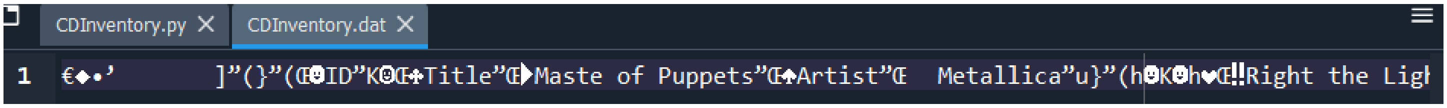
## Running the script in Spyder



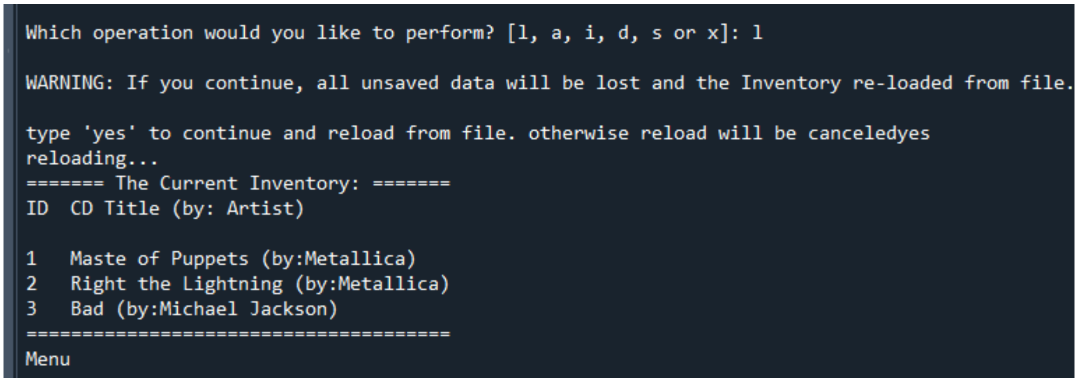
```
Which operation would you like to perform? [l, a, i, d, s or x]: a


Enter ID: 3

What is the CD's title? Bad

What is the Artist's name? Michael Jackson
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   Maste of Puppets (by:Metallica)
2   Right the Lightning (by:Metallica)
3   Bad (by:Michael Jackson)
======================================
```



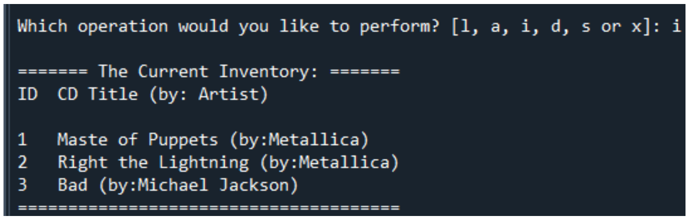
```
Which operation would you like to perform? [l, a, i, d, s or x]: s

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

1   Maste of Puppets (by:Metallica)
2   Right the Lightning (by:Metallica)
3   Bad (by:Michael Jackson)
======================================

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
```



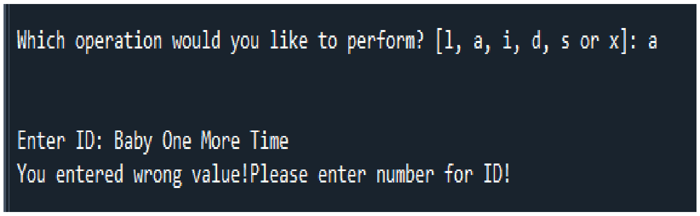
```
1  €◆•’        ]”(}”(Œ☻ID”K☺Œ✥Title”Œ▶Maste of Puppets”Œ▲Artist”Œ  Metallica”u}”(h☻K☻h♥Œ‼Right the Ligh
```



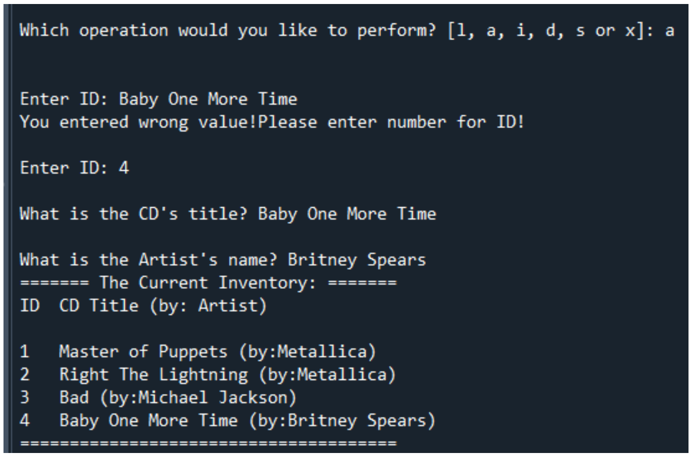
```
Which operation would you like to perform? [l, a, i, d, s or x]: l

WARNING: If you continue, all unsaved data will be lost and the Inventory re-loaded from file.

type 'yes' to continue and reload from file. otherwise reload will be canceledyes
reloading...
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   Maste of Puppets (by:Metallica)
2   Right the Lightning (by:Metallica)
3   Bad (by:Michael Jackson)
======================================
Menu
```



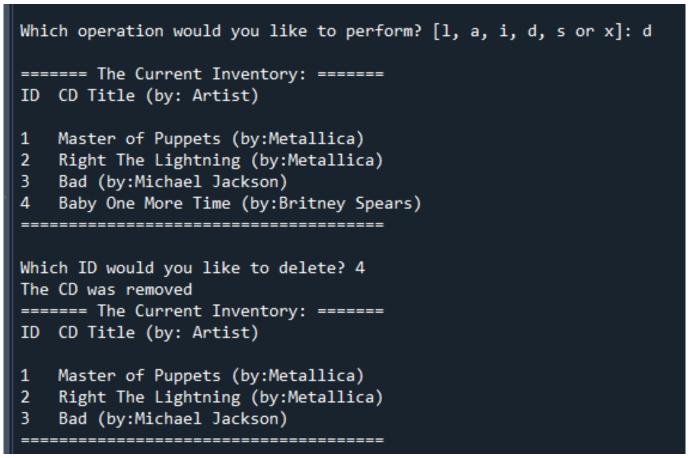
```
Which operation would you like to perform? [l, a, i, d, s or x]: i

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

1   Maste of Puppets (by:Metallica)
2   Right the Lightning (by:Metallica)
3   Bad (by:Michael Jackson)
======================================
```



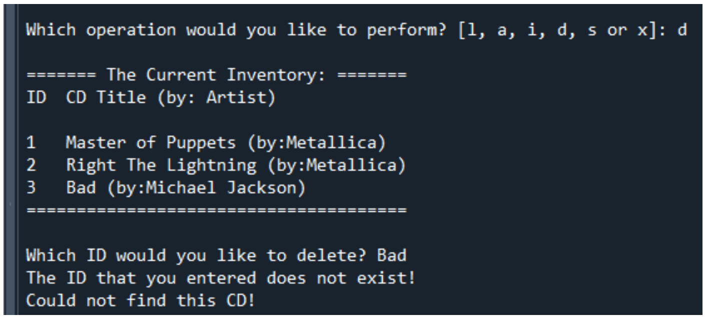
```
Which operation would you like to perform? [l, a, i, d, s or x]: a


Enter ID: Baby One More Time
You entered wrong value!Please enter number for ID!
```



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


Enter ID: Baby One More Time
You entered wrong value!Please enter number for ID!

Enter ID: 4

What is the CD's title? Baby One More Time

What is the Artist's name? Britney Spears
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   Master of Puppets (by:Metallica)
2   Right The Lightning (by:Metallica)
3   Bad (by:Michael Jackson)
4   Baby One More Time (by:Britney Spears)
======================================
```

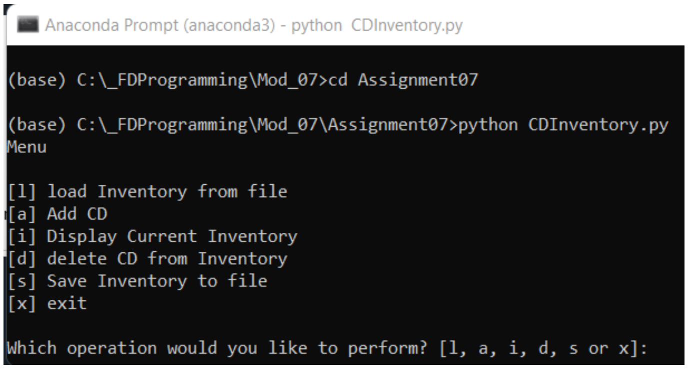


```
Which operation would you like to perform? [l, a, i, d, s or x]: d

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

1   Master of Puppets (by:Metallica)
2   Right The Lightning (by:Metallica)
3   Bad (by:Michael Jackson)
4   Baby One More Time (by:Britney Spears)
======================================

Which ID would you like to delete? 4
The CD was removed
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   Master of Puppets (by:Metallica)
2   Right The Lightning (by:Metallica)
3   Bad (by:Michael Jackson)
======================================
```

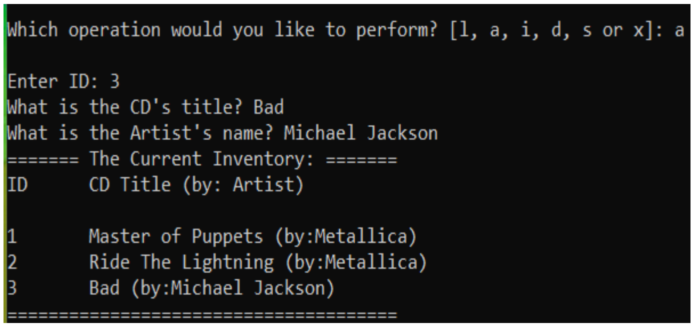


```
Which operation would you like to perform? [l, a, i, d, s or x]: d

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

1   Master of Puppets (by:Metallica)
2   Right The Lightning (by:Metallica)
3   Bad (by:Michael Jackson)
======================================

Which ID would you like to delete? Bad
The ID that you entered does not exist!
Could not find this CD!
```

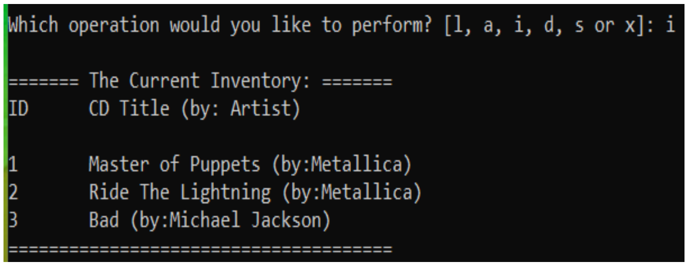
## Running the script in terminal window



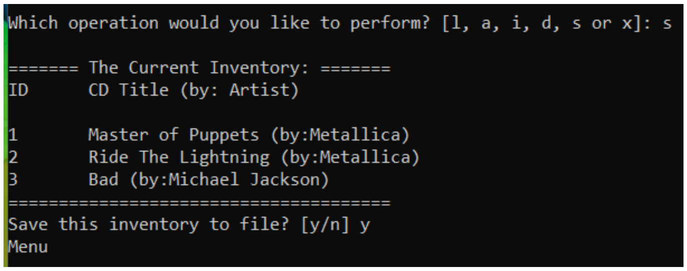
```
(base) C:\_FDProgramming\Mod_07>cd Assignment07

(base) C:\_FDProgramming\Mod_07\Assignment07>python CDInventory.py
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]:
```



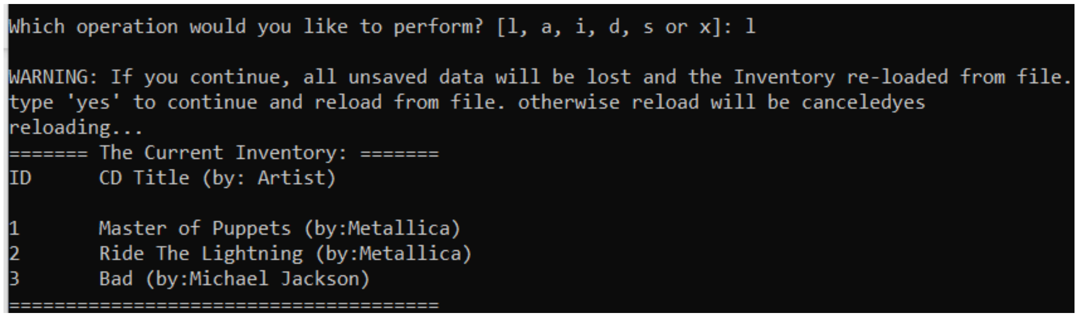
```
Which operation would you like to perform? [l, a, i, d, s or x]: a

Enter ID: 3
What is the CD's title? Bad
What is the Artist's name? Michael Jackson
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Master of Puppets (by:Metallica)
2       Ride The Lightning (by:Metallica)
3       Bad (by:Michael Jackson)
======================================
```



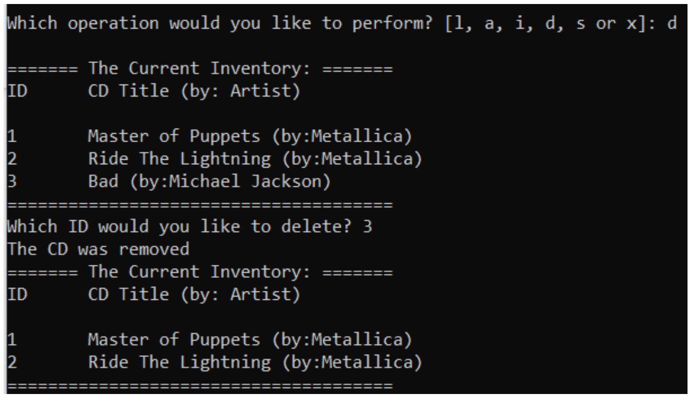
```
Which operation would you like to perform? [l, a, i, d, s or x]: i

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

1       Master of Puppets (by:Metallica)
2       Ride The Lightning (by:Metallica)
3       Bad (by:Michael Jackson)
======================================
```



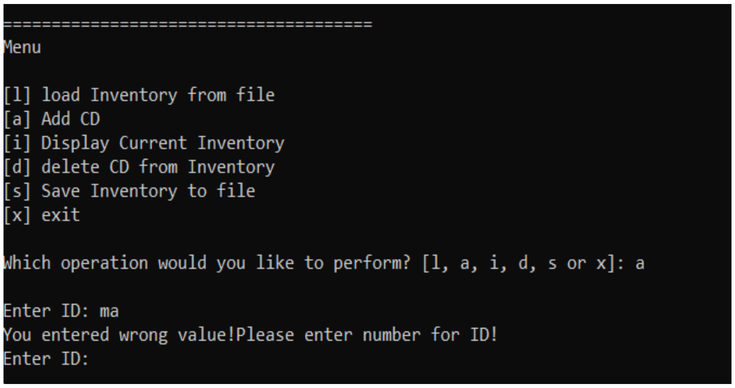
```
Which operation would you like to perform? [l, a, i, d, s or x]: s

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

1       Master of Puppets (by:Metallica)
2       Ride The Lightning (by:Metallica)
3       Bad (by:Michael Jackson)
======================================
Save this inventory to file? [y/n] y
Menu
```



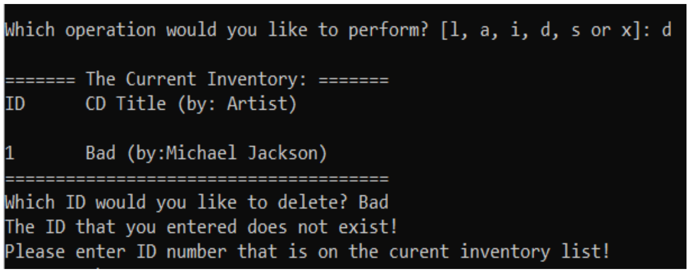
```
Which operation would you like to perform? [l, a, i, d, s or x]: l

WARNING: If you continue, all unsaved data will be lost and the Inventory re-loaded from file.
type 'yes' to continue and reload from file. otherwise reload will be canceledyes
reloading...
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Master of Puppets (by:Metallica)
2       Ride The Lightning (by:Metallica)
3       Bad (by:Michael Jackson)
======================================
```



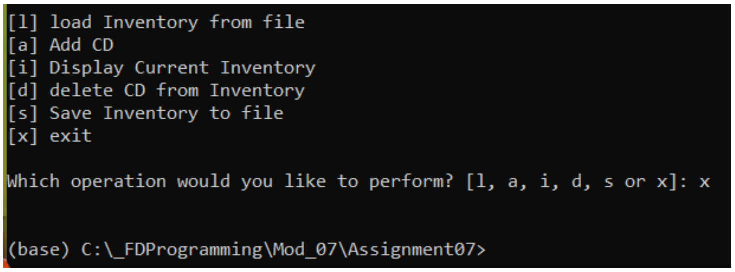
```
Which operation would you like to perform? [l, a, i, d, s or x]: d

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

1       Master of Puppets (by:Metallica)
2       Ride The Lightning (by:Metallica)
3       Bad (by:Michael Jackson)
======================================
Which ID would you like to delete? 3
The CD was removed
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Master of Puppets (by:Metallica)
2       Ride The Lightning (by:Metallica)
======================================
```



```
======================================
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 ID: ma
You entered wrong value!Please enter number for ID!
Enter ID:
```



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

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

1       Bad (by:Michael Jackson)
======================================
Which ID would you like to delete? Bad
The ID that you entered does not exist!
Please enter ID number that is on the curent inventory list!
```



```
[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


(base) C:\_FDProgramming\Mod_07\Assignment07>
```

## Summary

As we mentioned above, we cover exception handling and pickling in this module. Exception can be handled using a try statement. The critical operation which can raise an exception goes inside the try clause. We also mentioned that the process to convert python objects like lists and dictionaries into byte streams is called pickling. However, the process to convert byte streams back into python objects is called unpickling.

## Appendix

CDInventory.py using classes and functions, binary file and exception handling.

Link to my github.com repository for Assignment 07 https://github.com/Smileski/Assignment_07/