Brooke Biscoe
March 1, 2022
Foundations of Programming: Python
Assignment 07
https://github.com/Brocoe/IntroToProg-Python-Mod07
https://brocoe.github.io/IntroToProg-Python-Mod07/

Error Handling and Pickling

Introduction

This week I learnt about how to use built in python commands in customer functions and different type of file format. I also learnt some more advanced concepts on error handling and how to generate custom error messages to make things more readable. Finally, I reviewed the markdown language and GitHub and how to generate professional looking written documents.

Built-in Python Commands and Binary Files

One the key things that I learnt about this week was the advantages of putting built-in Python commands into custom functions. The key advantage to doing this is that you can build more customization and error handling in your own function versus compared to simply using the built in commands.

I also learned about a different kind of file which is a binary file. A binary file is a set of serialized data that obscure's the files content and potentially reduces size. Python does this using a technique known as pickling. Unlike a plain text file which requires that user simply store data to a file, when pickling a binary file, you can store any python object in there including lists, dictionaries or class objects. A full list of objects that can be pickled can be found on the following website https://www.datacamp.com/community/tutorials/pickle-python-tutorial. I learnt that pickling can be useful for data science to save models that have been trained rather than having to clean them up again. I also found some more information about pickling on the following website https://pythonprogramming.net/python-pickle-module-save-objects-serialization/

Error Handling

One of the major topics that I learnt about in this lecture was error handling . Specifically, I learnt that one of the key advantages to structured error handling is that you can handle common errors that a user a runs into and write custom messages that are more user friendly than the built-in python messages. Also, using a Try-Except block can prevent a code from crashing if there is an error.

The except class is a class in Python that provides information about errors that may appear in someone’s code. The exception class can be called in order to handle errors that may come up as a result of user error. You can call methods off of the exception class in order to make an error message more readable.

One can derive a new class from the exception class by creating a new class and passing ‘Exception’ as the only argument for the class. Then you can create a variety of methods to call similar to the built-in error handling methods. Below is snippet of some of my code from the assignment where I built in some custom error handling:

```
class InvalidOptionError(Exception):
    def __str__(self):
        global options
        return 'This is an invalid option. The available options are {}.'.format(options)
class NotUniqueEmployeeIDError(Exception):
    def __str__(self):
        return 'This employee ID is already in use. You must create a unique ID'
class MustNotBeNumericError(Exception):
    def __str__(self):
        return 'This variable MUST NOT contain numbers. Please enter it again.'
class MustBeNumericError(Exception):
    def __str__(self):
        return 'This variable MUST contain POSITIVE numbers. Please enter it again.'
class NoSearchResultError(Exception):
    def __str__(self):
        return 'There was no with data with this value in this column.\n' \
               'Try changing your search parameters'
class AvailableDepartmentsError(Exception):
    def __str__(self):
        global departmentNames
        return 'That is not a valid department. The available departments are {}.'.format(departmentNames)
class ValidNameError(Exception):
    def __str__(self):
        return 'Employee name must be entered as <First Name> <Last Name>, ' \
               'with only a single space between them.'
class AvailableColumnsError(Exception):
    def __str__(self):
        global keys
        return 'That is not a valid search field. The available fields are {}.'.format(keys)
```

It would be work creating a custom error message if there are specific errors that you think users of your program might make. I made a few custom error messages in my code. For example, the code below is something I created in order to handle an error that would occur if a user did not input a name correctly.

```
class ValidNameError(Exception):
    def __str__(self):
        return 'Employee name must be entered as <First Name> <Last Name>, ' \
               'with only a single space between them.'
```

Markdown Language

The markdown language is the language used by GitHub in order to format documents that are hosted on their website. The Markdown code is converted to HTML using a conversion process application called Jekyll. Since Markdown can be converted to HTML they can be used for GitHub webpages.

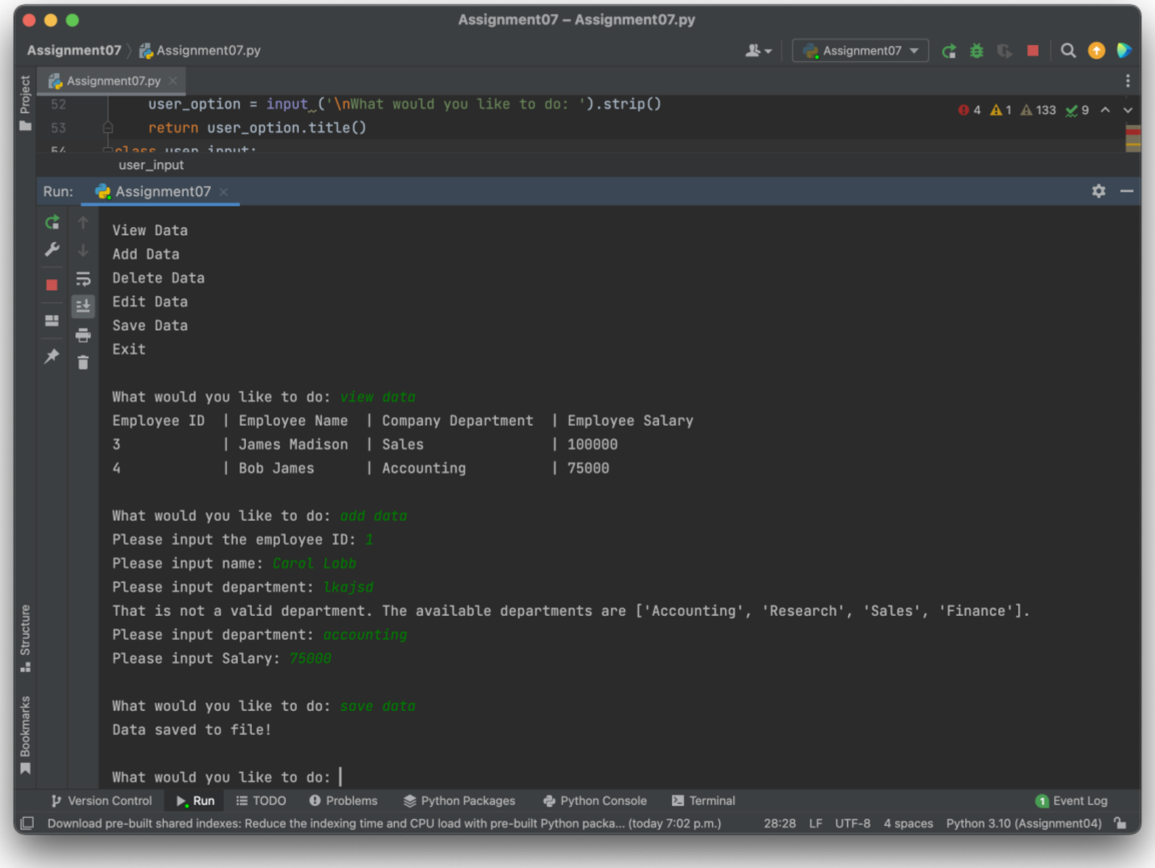
Assignment

In the assignment this week I had to demonstrate my ability to work with binary files and structured error handling. Admittedly, I went a bit overboard and ended up developing a whole program that stores and employee database in a binary file as well as handles multiple errors to ensure that data is stored consistently. Below is an example of some code I used whereby a user can search for data and remove it from the database (or more accurately not add it to a new database). The code includes a structured error handling which will return a message if the data that is searched for is not found.

```
def DeleteData(lst,key,search):
    new_lst = []
    try:
        cntr = 0
        for row in lst:
            if not row[key] == search:
                new_lst.append(row)
            else:
                cntr += 1
        if cntr == 0:
            raise NoSearchResultError
        print('Data Deleted')
    except NoSearchResultError as e:
        print(e)
    return new_lst
```

```
class NoSearchResultError(Exception):
  def __str__(self):
    return 'There was no with data with this value in this column.\n' \
        'Try changing your search parameters'
```

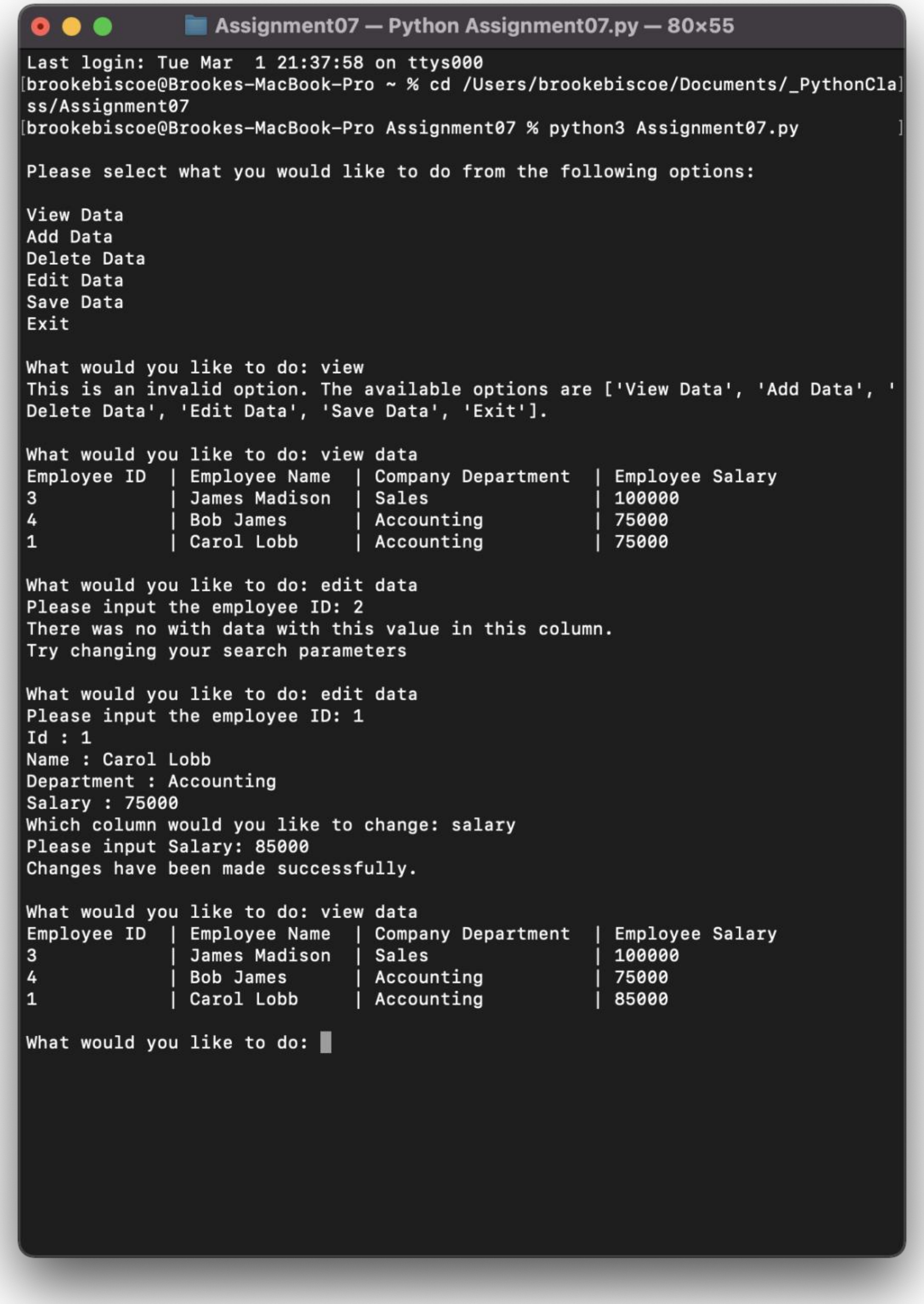
My code was able to be used in both PyCharm as well as the OS Command Shell as shown below.



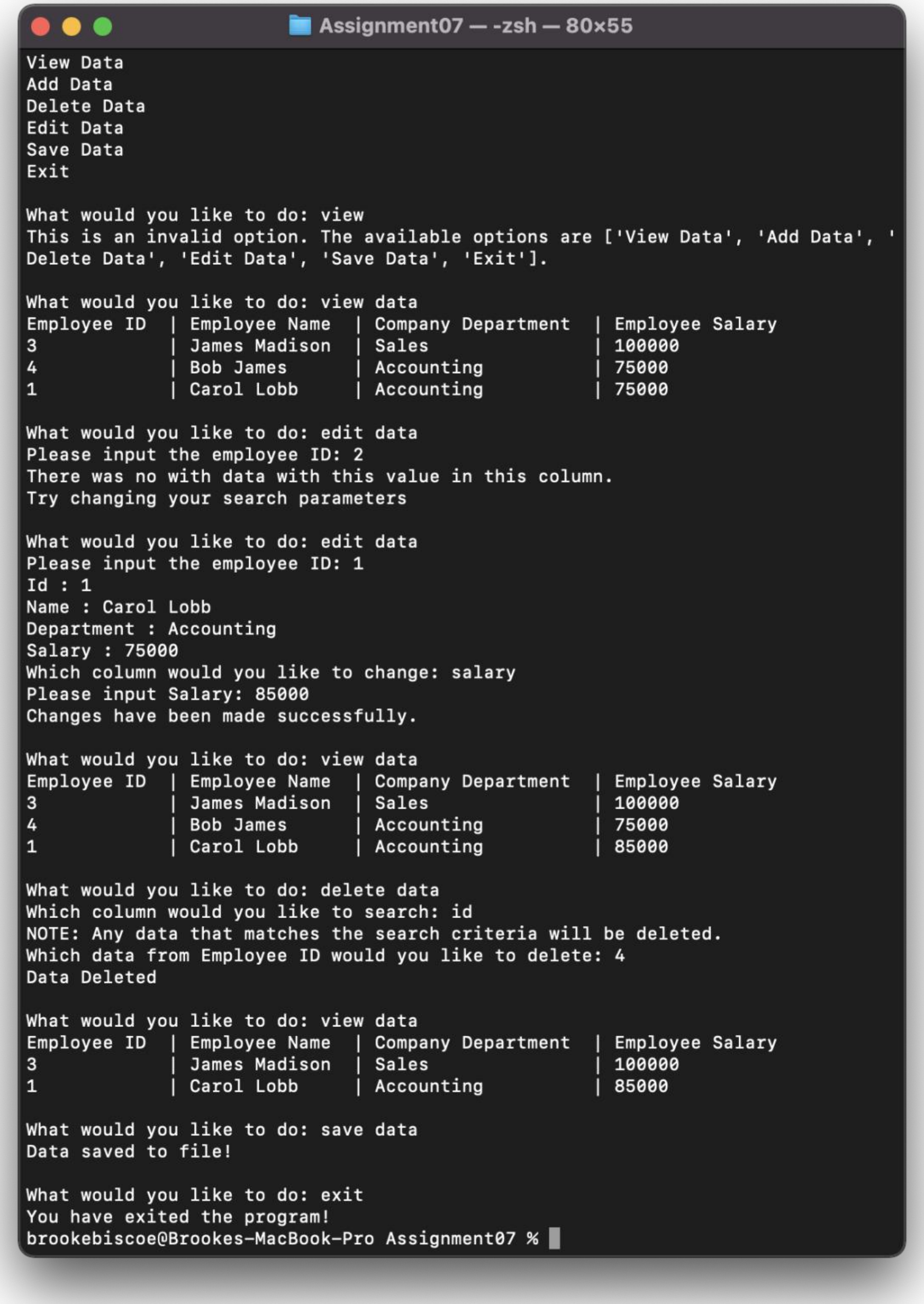
Assignment07 – Assignment07.py
Assignment07 〉 Assignment07.py
Assignment07
Project
Assignment07.py
4 1 133 9
user_input
Run: Assignment07
Structure
Bookmarks
Version Control
Run
TODO
Problems
Python Packages
Python Console
Terminal
Event Log
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packa... (today 7:02 p.m.)
28:28 LF UTF-8 4 spaces Python 3.10 (Assignment04)

```
        user_option = input('\nWhat would you like to do: ').strip()
        return user_option.title()
class user_input:
```

```
View Data
Add Data
Delete Data
Edit Data
Save Data
Exit

What would you like to do: view data
Employee ID  | Employee Name  | Company Department  | Employee Salary
3            | James Madison  | Sales               | 100000
4            | Bob James      | Accounting          | 75000

What would you like to do: add data
Please input the employee ID: 1
Please input name: Carol Lobb
Please input department: lkajsd
That is not a valid department. The available departments are ['Accounting', 'Research', 'Sales', 'Finance'].
Please input department: accounting
Please input Salary: 75000

What would you like to do: save data
Data saved to file!

What would you like to do: 
```



```
Last login: Tue Mar  1 21:37:58 on ttys000
[brookebiscoe@Brookes-MacBook-Pro ~ % cd /Users/brookebiscoe/Documents/_PythonCla]
ss/Assignment07
[brookebiscoe@Brookes-MacBook-Pro Assignment07 % python3 Assignment07.py         ]

Please select what you would like to do from the following options:

View Data
Add Data
Delete Data
Edit Data
Save Data
Exit

What would you like to do: view
This is an invalid option. The available options are ['View Data', 'Add Data', '
Delete Data', 'Edit Data', 'Save Data', 'Exit'].

What would you like to do: view data
Employee ID  | Employee Name  | Company Department  | Employee Salary
3            | James Madison  | Sales               | 100000
4            | Bob James      | Accounting          | 75000
1            | Carol Lobb     | Accounting          | 75000

What would you like to do: edit data
Please input the employee ID: 2
There was no with data with this value in this column.
Try changing your search parameters

What would you like to do: edit data
Please input the employee ID: 1
Id : 1
Name : Carol Lobb
Department : Accounting
Salary : 75000
Which column would you like to change: salary
Please input Salary: 85000
Changes have been made successfully.

What would you like to do: view data
Employee ID  | Employee Name  | Company Department  | Employee Salary
3            | James Madison  | Sales               | 100000
4            | Bob James      | Accounting          | 75000
1            | Carol Lobb     | Accounting          | 85000

What would you like to do: 
```



```
View Data
Add Data
Delete Data
Edit Data
Save Data
Exit

What would you like to do: view
This is an invalid option. The available options are ['View Data', 'Add Data', '
Delete Data', 'Edit Data', 'Save Data', 'Exit'].

What would you like to do: view data
Employee ID  | Employee Name  | Company Department  | Employee Salary
3            | James Madison  | Sales               | 100000
4            | Bob James      | Accounting          | 75000
1            | Carol Lobb     | Accounting          | 75000

What would you like to do: edit data
Please input the employee ID: 2
There was no with data with this value in this column.
Try changing your search parameters

What would you like to do: edit data
Please input the employee ID: 1
Id : 1
Name : Carol Lobb
Department : Accounting
Salary : 75000
Which column would you like to change: salary
Please input Salary: 85000
Changes have been made successfully.

What would you like to do: view data
Employee ID  | Employee Name  | Company Department  | Employee Salary
3            | James Madison  | Sales               | 100000
4            | Bob James      | Accounting          | 75000
1            | Carol Lobb     | Accounting          | 85000

What would you like to do: delete data
Which column would you like to search: id
NOTE: Any data that matches the search criteria will be deleted.
Which data from Employee ID would you like to delete: 4
Data Deleted

What would you like to do: view data
Employee ID  | Employee Name  | Company Department  | Employee Salary
3            | James Madison  | Sales               | 100000
1            | Carol Lobb     | Accounting          | 85000

What would you like to do: save data
Data saved to file!

What would you like to do: exit
You have exited the program!
brookebiscoe@Brookes-MacBook-Pro Assignment07 % 
```

Conclusion

This week I learnt about binary files and how they are different from plain texts files. I learnt how to create custom error messages and how structured error handling can really improve my code. Finally, I learned about the markdown language and how to use it create a a very nice GitHub Web Page.