**Python References**

**OOPs**

* https://realpython.com/python3-object-oriented-programming/

**Closures**

* https://www.pythontutorial.net/advanced-python/python-closures/

**Decorator**

* https://realpython.com/primer-on-python-decorators/

**Generator**

* https://realpython.com/introduction-to-python-generators/

**Iterator**

* https://realpython.com/python-iterators-iterables/

**Exception handling**

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

**File I/O operations**

* https://pynative.com/python/file-handling/

**With statement**

* https://realpython.com/python-with-statement/

**Modules**

* https://realpython.com/python-modules-packages/

**Unit Testing using PyTest**

* https://docs.pytest.org/en/stable/getting-started.html

**Accessing Excel/CSV/JSON using Python libraries**

* https://openpyxl.readthedocs.io/en/stable/tutorial.html
* https://realpython.com/python-csv
* https://realpython.com/python-json

**Accessing MySQL database using Python**

* https://realpython.com/python-mysql/
* https://www.datacamp.com/tutorial/mysql-python

**Code Quality ,Debugging**

* https://www.jetbrains.com/help/pycharm/part-1-debugging-python-code.html
* https://peps.python.org/pep-0008/
* <https://pypi.org/project/pylint/>

**Demo URL**

* https://github.com/Duraiatpeers/nessdemosjul29

**Practice quiz**

* <https://forms.gle/9MsFhW2PKpqp4jkn8>

**Practice Exercises**

**Instructions:**

1. Go through the exercises and write the solutions.
2. Use OOPs concepts for your implementation
3. You can go with assumptions if not clear on any specific problem and have your implementations.
4. As and when you complete exercises, push it to GitHub and share it with me ( GitHub id: duraiatpeers)
5. Your code will be reviewed, and the feedback will be shared

**Exercise 1:**

**Problem statement:**

    1. create a data file data.txt with the following data

    empid,empname,emplocation,empsalary

    e001,iniyal,chennai,20000.00

    e002,aniyal,bangalore,25000.00

    e003,indulekha,trivandrum,18000.00

    2. open the file, read line by line

    - extract data and put it in the following format

            {

            "employees": [

                    {

                        "empid":"e001",

                        "empname":"iniyal",

                        "emplocation":"chennai"

                        "empsalary":20000.00

                    },

                    {

                        "empid":"e002",

                        "empname":"iniyal",

                        "emplocation":"chennai"

                        "empsalary":20000.00

                    },{

                    }

                ]

            }

**Problem statement:**

    create a data file data.txt with the following data

    empid,empname,emplocation,empsalary

    e001,iniyal,chennai,20000.00

    e002,aniyal,bangalore,25000.00

    1. Get the header columns from the user separated by ,

    For e.g  empid,empname

            empname

            empid,empname,emplocation

            \*

            empname,empsalary

    Output should be based on the header columns

    For e.g, if the userinput is empname, empsalary, then the following data should be printed

    2. open the file, read line by line

        - extract data and put it in the following format

            {

            "employees": [

                    {

                        "empname":"iniyal",

                        "empsalary":20000.00

                    },

                    {

                        "empname":"iniyal",

                        "empsalary":20000.00

                    }

                ]

            }

    Output format:

    dict

    - list

        - dict

**Exercise 2**

        Create a child class Bus that will inherit all of the variables and methods of the Vehicle

            class

        Given:

        class Vehicle:

        def \_\_init\_\_(self, name, max\_speed, mileage):

            self.name = name

            self.max\_speed = max\_speed

            self.mileage = mileage

        Create a Bus object that will inherit all of the variables and methods of the parent Vehicle

            class and display it.

        Expected Output:

        Vehicle Name: School Volvo Speed: 180 Mileage: 12

**Exercise 3**

        Create a Bus child class that inherits from the Vehicle class. The default fare charge of

        any vehicle is seating capacity \* 100. If Vehicle is Bus instance, we need to add an extra

        10% on full fare as a maintenance charge. So total fare for bus instance will become the

        final amount = total fare + 10% of the total fare.

        Note: The bus seating capacity is 50. so the final fare amount should be 5500. You need to

        override the fare() method of a Vehicle class in Bus class.

        class Vehicle:

            def \_\_init\_\_(self, name, mileage, capacity):

                self.name = name

                self.mileage = mileage

                self.capacity = capacity

            def fare(self):

                return self.capacity \* 100

**Exercise 4:**

* Write a program to read a file containing multiple sentences. Get words from those sentences and store it in a string list. Then find the count of the number of words whose length is odd number. The method should take the list array as input and should return an integer which is the count of the number of words with length being odd number. The method should return -1 if the list is empty.
* Test the solution using pytest.
* Concepts to be used ( not limited to): Class, Objects, methods, File handling, Exception Handling, Pytest