USE CASES

1) **Write a Python program which accepts a list named : randomList = ['a', 0, 2]. Use exception handling using try-catch which gives the output as:**

Output:

1) If the List element is a alphabet or string, the output will be
The entry is a

Oops! <class
'ValueError'>occured. Next entry.

2) If the List element is “0”,the output will be
The entry is 0
Oops! <class '
ZeroDivisionError’>occured. Next entry.

3) If the List element is and integer except 0,then output will be
The entry is 2

The reciprocal of 2 is 0.5 // reciprocal of an integer

2) **Array out of Bound Exception**

Write a Python program to give exception “Array Out of Bound” if the user wants to access the elements beyond the list size (use try and except)

3) Write a python module script that contains fib2() method to calculate the Fibonacci series till 1000 and save it as fibo.py.

Note : The module created as fibo.py has to be placed in lib folder:

For linux/ubuntu path = /home/anaconda/lib/python3
For Windows path = C:\Users\Acadgild\Anaconda3\Lib

4) Write a python module script that contains ispalindrome() method to calculate the input string as palindrome string or not and save it as palindrome.py.

5) Write a program in Python with one class called Cipher. Within the constructor of this class, ask user for a string and store it. Use a static variable, key to store a randomly generated integer between 1 and 50 inclusive. Implement two methods, encrypt and decrypt within this class. Encrypt generates and prints a cipher text using the user-entered string and the key and ecrypt generates decrypted string from ciphertext. The cipher only encrypts alpha and numeric (A-Z, a-z, 0-9). All Symbols, such as - , ; %, remain unencrypted. The cipher text can have special characters. Use generator expression to filter out alpha and numeric characters of the input string and to generate cipher text. Create an instance of this class, encrypt and decrypt back the user entered string.

6) Get Data from the following link:

http://files.grouplens.org/datasets/movielens/ml-20m.zip

We will be using the following files for this exercise:

ratings.csv : userId,movieId,rating, timestamp
tags.csv : userId,movieId, tag, timestamp
movies.csv : movieId, title, genres

I. **Read the dataset using pandas.**
II. **Extract the first row from tags and print its type.**
III. **Extract row 0, 11, 2000 from tags DataFrame.**

V. **Calculate descriptive statistics for the 'ratings' column of the ratings DataFrame. Verify using describe().**
VI. **Filter out ratings with rating > 5**
VII. **Find how many null values, missing values are present. Deal with them. Print out how many rows have been modified.**
VIII. **Filter out movies from the movies DataFrame that are of type 'Animation'.**
IX. **Find the average rating of movies.**
X. **Perform an inner join of movies and tags based on movieId.**
XI. **Print out the 5 movies that belong to the Comedy genre and have rating greater than 4.**
XII. **Split 'genres' into multiple columns.**
XIII. **Extract year from title e.g. (1995).**
XIV. **Select rows based on timestamps later than 2015-02-01.**
XV. **Sort the tags DataFrame based on timestamp.**

Expected Output: This assignment consists of 500 marks and has to be submitted in .ipynb/PDF format in the upcoming session for evaluation.