Bharath Kumar N

PS ID:10843180

Python Mock assessment 1

Linux 1.2

PROGRAM 3:

File Handling Utility – Text Analyzer Objective: Read sample .txt file and display all Unique cities sample.txt (copy the below to sample.txt file)

EmpID,Name,Department,Location

E1001,Asha Rao,Data Science,Mumbai

E1002,Rahul Mehta,IT Support,Hyderabad

E1003,Neha Singh,Human Resources,Hyderabad

E1004,Vikram Das,Finance,Mumbai

E1005,Priya Kapoor,Marketing,Hyderabad

Requirements:

λ Ask the user for a file path; open safely with try/except for FileNotFoundError.

λ Use if condition whereever required

λ Use user defined function

Sample Input/Output:

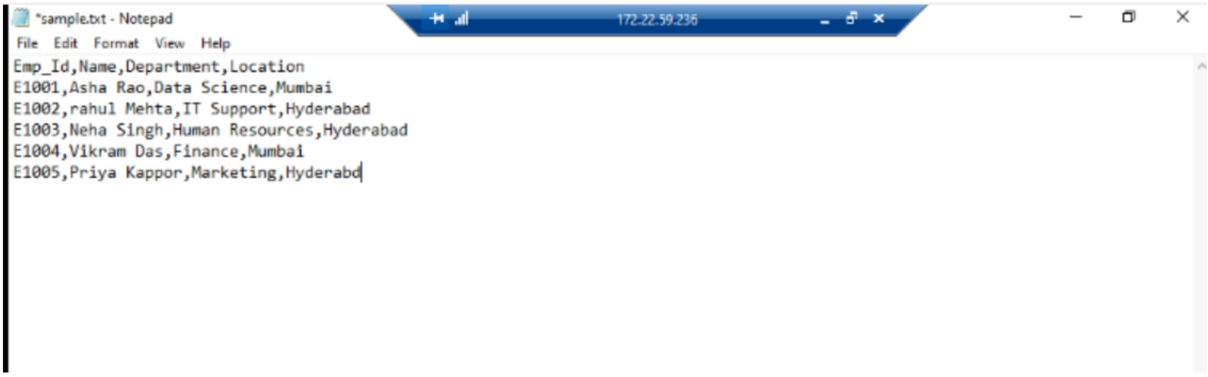
Sample Input:

Enter path to a .txt file: sample.txt

Sample Output: < all unique city names line by line >

Answer

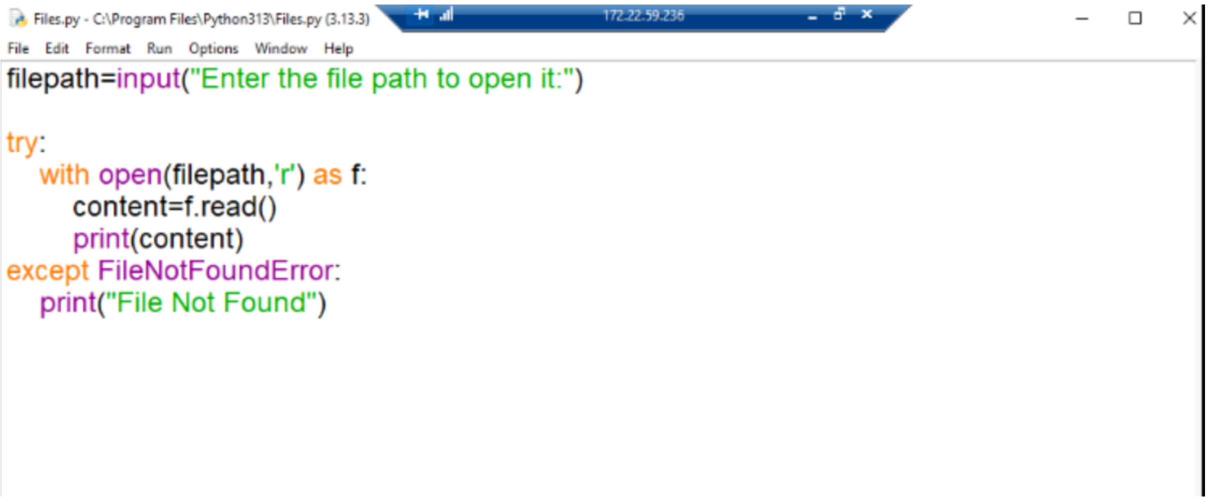
Step 1: Creating A Sample.txt document



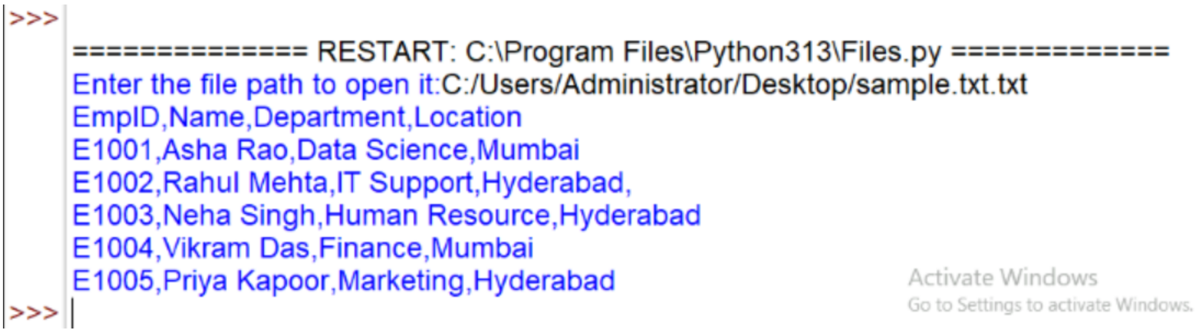
```
Emp_Id,Name,Department,Location
E1001,Asha Rao,Data Science,Mumbai
E1002,rahul Mehta,IT Support,Hyderabad
E1003,Neha Singh,Human Resources,Hyderabad
E1004,Vikram Das,Finance,Mumbai
E1005,Priya Kappor,Marketing,Hyderabd
```

Step 2: Including try .. except concept for file not found error

And using read method trying to read the file

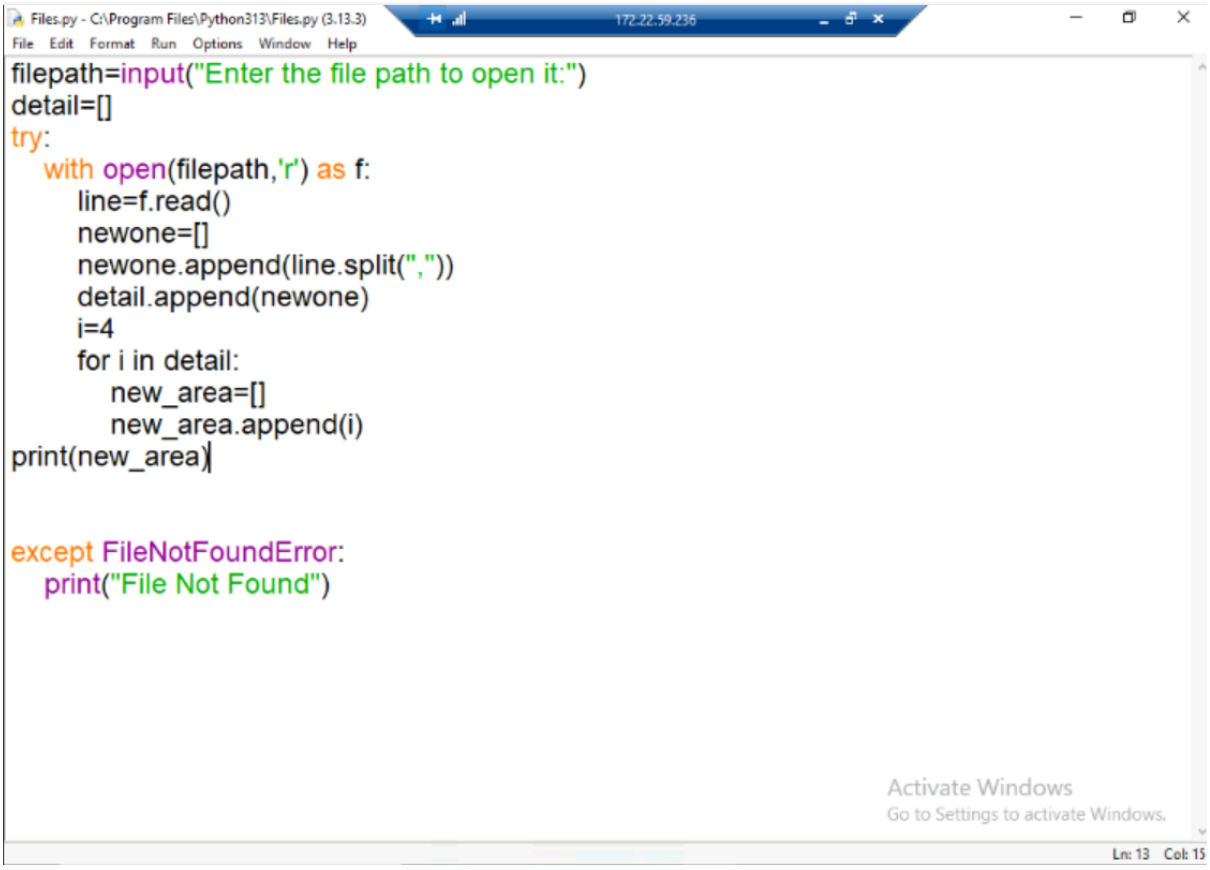

Files.py - C:\Program Files\Python313\Files.py (3.13.3)
172.22.59.236
File Edit Format Run Options Window Help

```python
filepath=input("Enter the file path to open it:")

try:
    with open(filepath,'r') as f:
        content=f.read()
        print(content)
except FileNotFoundError:
    print("File Not Found")
```



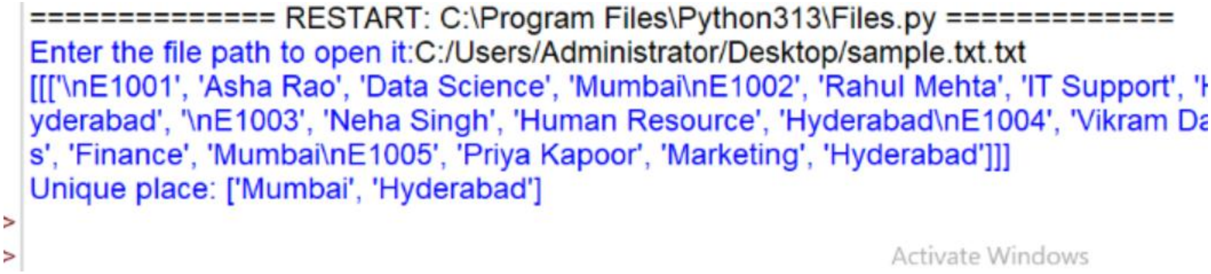
```
>>>
    ============= RESTART: C:\Program Files\Python313\Files.py ============
    Enter the file path to open it:C:/Users/Administrator/Desktop/sample.txt.txt
    EmpID,Name,Department,Location
    E1001,Asha Rao,Data Science,Mumbai
    E1002,Rahul Mehta,IT Support,Hyderabad,
    E1003,Neha Singh,Human Resource,Hyderabad
    E1004,Vikram Das,Finance,Mumbai
    E1005,Priya Kapoor,Marketing,Hyderabad
>>>
```

Step 3: Find and iterate the location details


Files.py - C:\Program Files\Python313\Files.py (3.13.3)
172.22.59.236
File Edit Format Run Options Window Help
Activate Windows
Go to Settings to activate Windows.
Ln: 13 Col: 15

```python
filepath=input("Enter the file path to open it:")
detail=[]
try:
    with open(filepath,'r') as f:
        line=f.read()
        newone=[]
        newone.append(line.split(","))
        detail.append(newone)
        i=4
        for i in detail:
            new_area=[]
            new_area.append(i)
print(new_area)


except FileNotFoundError:
    print("File Not Found")
```

Output:



```
============== RESTART: C:\Program Files\Python313\Files.py =============
Enter the file path to open it:C:/Users/Administrator/Desktop/sample.txt.txt
[[['\nE1001', 'Asha Rao', 'Data Science', 'Mumbai\nE1002', 'Rahul Mehta', 'IT Support', 'H
yderabad', '\nE1003', 'Neha Singh', 'Human Resource', 'Hyderabad\nE1004', 'Vikram Da
s', 'Finance', 'Mumbai\nE1005', 'Priya Kapoor', 'Marketing', 'Hyderabad']]]
Unique place: ['Mumbai', 'Hyderabad']
>
>
```

PROGRAM 4: :

List Methods — Clean & Normalize Shopping List Objective:

Use only list methods (append, extend, insert, remove, pop, index, count, sort, reverse, slicing).

Requirements:

λ Input: a raw list like [" Milk", "eggs", "MILK ", "bread", "Eggs", " butter "].

λ Trim whitespace, convert to lowercase, and remove duplicates while preserving order (no set).

λ Print final sorted list and also reversed order.

Sample Input/Output:

Sample Input: [" Milk", "eggs", "MILK ", "bread", "Eggs", " butter "]

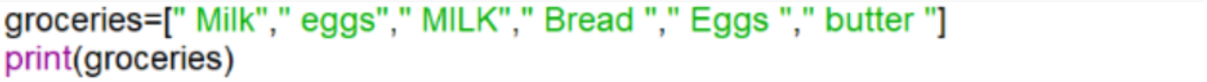
Sample Output: Counts: milk: 2, eggs: 2, bread: 1, butter: 1

Clean list (sorted): ['bread', 'butter', 'eggs', 'milk']

Reversed: ['milk', 'eggs', 'butter', 'bread'

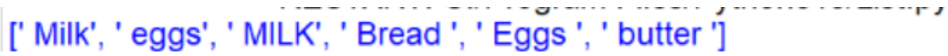
Answer

Step 1: A list is created



```
groceries=[" Milk"," eggs"," MILK"," Bread "," Eggs "," butter "]
print(groceries)
```

Output :



```
[' Milk', ' eggs', ' MILK', ' Bread ', ' Eggs ', ' butter ']
```

Step 2:

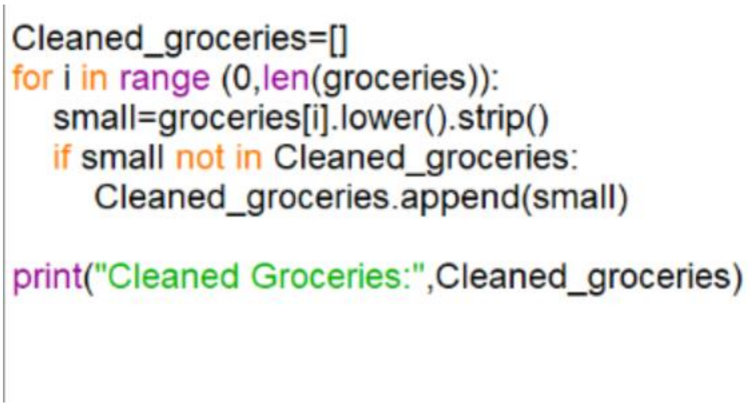
Lets create a empty grocery list

Using loops we iterate each and every thing in the grocery list

Using lower() function to change it into lower case

Using strip function to trim the white spaces

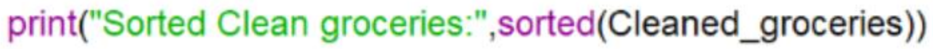
Then using conditionals we added the unique elements to the new grocery list



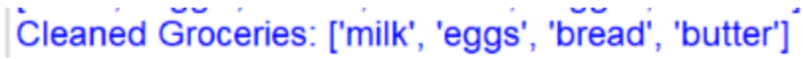
```
Cleaned_groceries=[]
for i in range (0,len(groceries)):
    small=groceries[i].lower().strip()
    if small not in Cleaned_groceries:
        Cleaned_groceries.append(small)

print("Cleaned Groceries:",Cleaned_groceries)
```

Step 3:

To sort the Cleaned grocery list use sorted method



```
print("Sorted Clean groceries:",sorted(Cleaned_groceries))
```

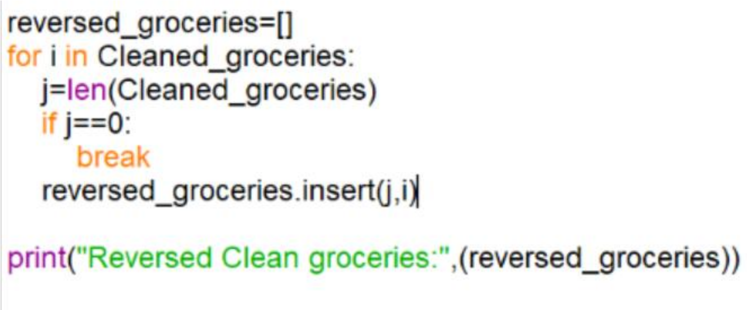


```
Cleaned Groceries: ['milk', 'eggs', 'bread', 'butter']
```

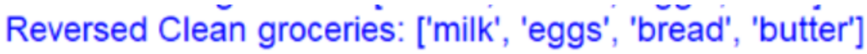
Step 4:

Inorder to reverse the grocery list we use a new empty list

We iterate through each list and insert it at the reverse proportional value of the index

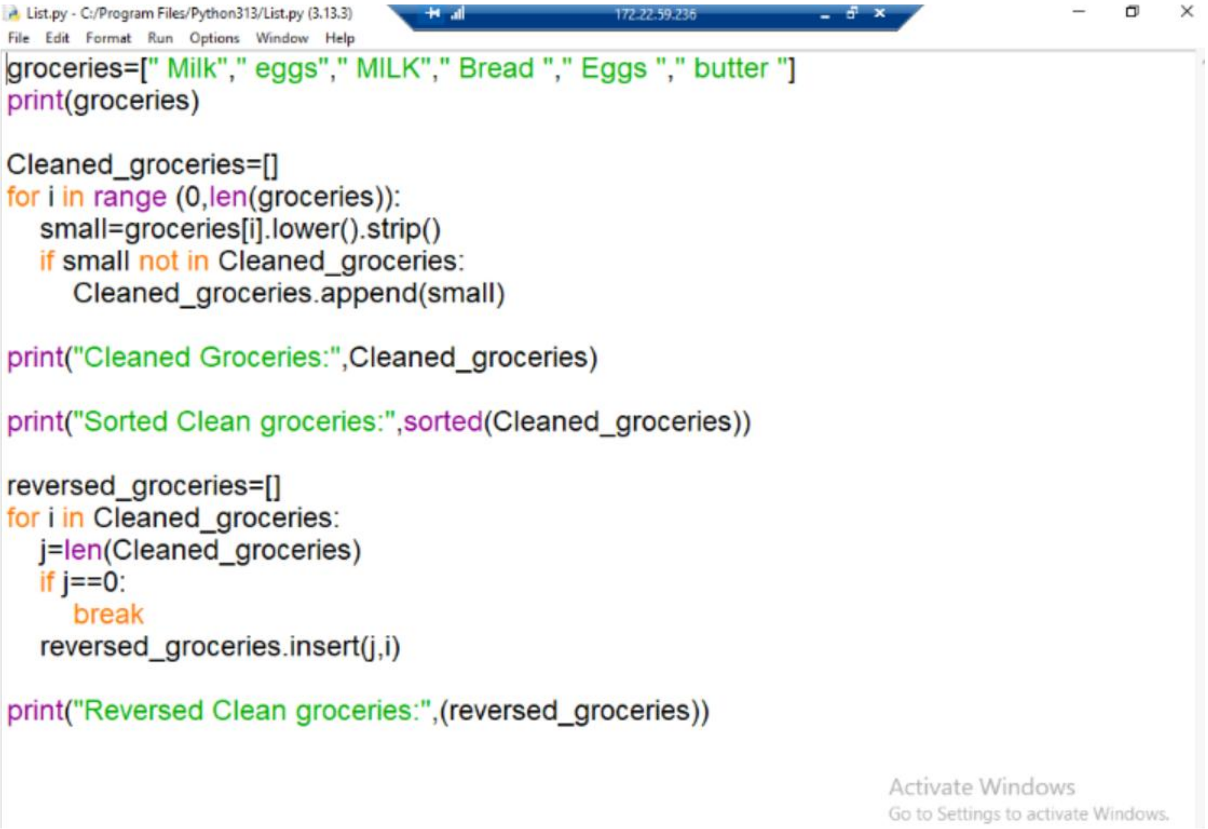


```
reversed_groceries=[]
for i in Cleaned_groceries:
    j=len(Cleaned_groceries)
    if j==0:
        break
    reversed_groceries.insert(j,i)

print("Reversed Clean groceries:",(reversed_groceries))
```



```
Reversed Clean groceries: ['milk', 'eggs', 'bread', 'butter']
```

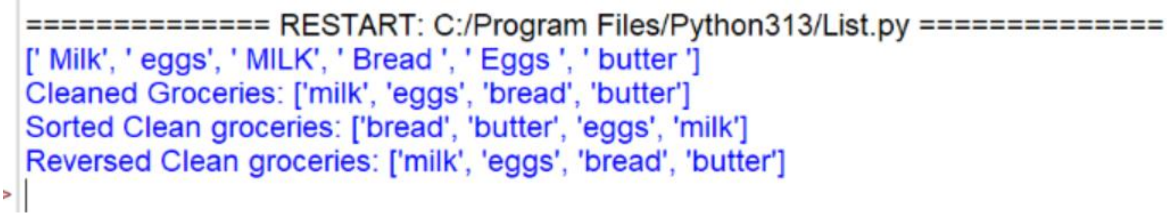
Overall Coding:



List.py - C:/Program Files/Python313/List.py (3.13.3)
172.22.59.236
File
Edit
Format
Run
Options
Window
Help
Activate Windows
Go to Settings to activate Windows.

```
groceries=[" Milk"," eggs"," MILK"," Bread "," Eggs "," butter "]
print(groceries)

Cleaned_groceries=[]
for i in range (0,len(groceries)):
    small=groceries[i].lower().strip()
    if small not in Cleaned_groceries:
        Cleaned_groceries.append(small)

print("Cleaned Groceries:",Cleaned_groceries)

print("Sorted Clean groceries:",sorted(Cleaned_groceries))

reversed_groceries=[]
for i in Cleaned_groceries:
    j=len(Cleaned_groceries)
    if j==0:
        break
    reversed_groceries.insert(j,i)

print("Reversed Clean groceries:",(reversed_groceries))
```

Overall Output:



```
============== RESTART: C:/Program Files/Python313/List.py ==============
[' Milk', ' eggs', ' MILK', ' Bread ', ' Eggs ', ' butter ']
Cleaned Groceries: ['milk', 'eggs', 'bread', 'butter']
Sorted Clean groceries: ['bread', 'butter', 'eggs', 'milk']
Reversed Clean groceries: ['milk', 'eggs', 'bread', 'butter']
>
```