File Handling and Error handling in **python**

File operations

```
#opening a File
file = open("/content/manjunath.txt", "r")

file = open("/content/manjunath.txt", "w")
print(file.write("this a command"))
file.close()
```

```
14
```

```
file = open("/content/manjunath.txt", "r")
conent = file.read()
print(conent)
file.close()
```

```
this a command
```

```
#Writing to a file
file = open("/content/manjunath.txt", "w")
print(file.write("jhfhf ihdfh jhkdfh hjskd"))
file.close()
```

```
24
```

```
#Appending to a File
file = open("/content/manjunath.txt", "a")
print(file.write("ram is a good programmer.\n"))
file.close()
```

```
26
```

```
#using with statement
with open("/content/manjunath.txt", "r") as file:
  content = file.read()
  print(content)
```

```
jhfhf ihdfh jhkdfh hjskdram is a good programmer.
```

```
#file Handling Mods
with open("/content/download.jpg", "rb") as file:
  data = file.read()
```

error **handling**

```
#Try Except Block
try:
  num = int(input("Enter a number:"))
  print(10/num)
except ZeroDivisionError:
  print("You cannot divide by zero.")
except ValueError:
  print("Invalid input. Please enter a valid number.")
```

```
Enter a number:0
You cannot divide by zero.
```

```
#Finally Block
try:
  file = open("example.text","r")
except FileNotFoundError:
  print("File not found.")
finally:
  print("Execution complete.")
```

```
File not found.
Execution complete.
```

```
#Raising Exceptions
def check_age(age):
  if age < 18:
    raise ValueError("Age must be 18 or older.")
    return True
    try:
      check_age(16)
    except ValueError as e:
      print(e)
```

Hands on practice

```
#Reading and Writing to a File
with open("/content/manjunath.txt", "w") as file:
  file.write("this is a my sample file")

#Appending data to a File
with open("/content/manjunath.txt", "a") as file:
  file.write("Let's Leran python./n")
```

```
Let's Leran python./n
```

```
with open("/content/manjunath.txt", "r") as file:
    print(file.read())
```

```
Let's Leran python./nLet's Leran python./n
```

```
#Handling Division by Zero Error
try:
  num1 = int(input("Enter a numerator: "))
  num2 = int(input("Enter a denominator: "))
  result = num1 / num2
  print("Result:",result)
except ZeroDivisionError:
    print("Cannot divide by zero.")
except ValueError:
    print("Invalid input! Enter numeric values.")
```

```
Enter a numerator: 5
Enter a denominator: 5
Result: 1.0
```

Problem ***solving***

```
#Create and Write to a File
with open("/content/manjunath.txt", "w") as file:
    file.write("Hello, World!")

#Read from a File
with open("/content/manjunath.txt", "r") as file:
    content = file.read()
    print(content)
```

```
Hello, World!
```

```
#Append to a File
with open("/content/manjunath.txt", "a") as file:
    file.write("\nWelcome to Python programming!")



#Generate a random number between 1 and 2
import random
random_number = random.randint(1,6)
print("the random number is:",random_number)
```

```
the random number is: 3
```

```
#Count Lines in a File
try:
  with open("/content/manjunath.txt", "r") as file:
    lines = file.readlines()
    print(f"Number of lines: {len(lines)}")
except FileNotFoundError:
  print("The file '/content/manjunath.txt' was not found in the
```

```
current directory.")
  print("Please make sure the file exists and the path is correct.")
```

```
Number of lines: 2
```

```
#Count Words in a File
with open("/content/manjunath.txt", "r") as file:
    words = file.read().split()
    print(f"Number of words: {len(words)}")
```

```
Number of words: 6
```

```
#Copy File Contents
with open("/content/manjunath.txt", "r") as source,
open("/content/manjunath.txt", "w") as destination:
    destination.write(source.read())

#Check if File Exists
import os
if os.path.exists("/content/manjunath.txt"):
    print("File exists")
else:
    print("File does not exist")
```

```
File exists
```

```
#Read File Line by Line
with open("/content/manjunath.txt", "r") as file:
    for line in file:
        print(line.strip())

#Search for a Word in a File
with open("/content/manjunath.txt", "r") as file:
    for i, line in enumerate(file, 1):
        if "Python" in line:
            print(f"Found on line {i}: {line.strip()}")

#Write a List to a File
numbers = [1, 2, 3, 4, 5]
with open("/content/manjunath.txt", "w") as file:
    file.write("\n".join(map(str, numbers)))

#Reverse File Contents
with open("/content/manjunath.txt", "r") as file:
    content = file.read()
with open("/content/manjunath.txt", "w") as file:
    file.write(content[::-1])


#File Statistics
with open("/content/manjunath.txt", "r") as file:
```

```
    content = file.read()
    lines = content.splitlines()
    words = content.split()
    characters = len(content)
    print(f"Characters: {characters}, Words: {len(words)}, Lines:
{len(lines)}")
```

```
Characters: 9, Words: 5, Lines: 5
```

```
#Merge Two Files
with open("/content/manjunath.txt", "r") as file1,
open("/content/manjunath.txt", "r") as file2,
open("/content/manjunath.txt", "w") as merged:
    merged.write(file1.read() + "\n" + file2.read())
```

```
#Count Occurrences of a Word
with open("/content/manjunath.txt", "r") as file:
    content = file.read()
    count = content.count("Python")
    print(f"'Python' appears {count} times")
```

```
'Python' appears 0 times
```

```
#Remove a Word from a File
with open("/content/manjunath.txt", "r") as file:
    content = file.read().replace("Hello", "")
with open("data.txt", "w") as file:
    file.write(content)
```

```
#File Encryption
with open("data.txt", "r") as file:
    content = file.read()
encrypted = ''.join(chr(ord(c) + 2) for c in content)
with open("encrypted.txt", "w") as file:
    file.write(encrypted)
```

```
#File Decryption
with open("encrypted.txt", "r") as file:
    encrypted_content = file.read()
decrypted = ''.join(chr(ord(c) - 2) for c in encrypted_content)
print(decrypted)
```

```
#Remove Blank Lines
with open("data.txt", "r") as file:
    lines = [line for line in file if line.strip()]
```

```
with open("data.txt", "w") as file:
    file.writelines(lines)

#Find Longest Word in a File
with open("/content/manjunath.txt", "r") as file:
    words = file.read().split()

if words:
    longest_word = max(words, key=len)
    print(f"The longest word is: {longest_word}")
else:
    print("The file is empty or contains no words.")
```

```
The file is empty or contains no words.
```

```
#Word Frequency Analysis
from collections import Counter
with open("data.txt", "r") as file:
    words = file.read().split()
frequency = Counter(words)
print(frequency)
```

```
Counter()
```

Error Handling **problems**

```
#Handle File Not Found Error
try:
    with open("data.txt", "r") as file:
        print(file.read())
except FileNotFoundError:
    print("File not found")



#Handle Division by Zero
try:
    a = int(input("Enter numerator: "))
    b = int(input("Enter denominator: "))
    print(a / b)
except ZeroDivisionError:
    print("Cannot divide by zero")
```

```
Enter numerator: 56
Enter denominator: 56
1.0
```

```
#Invalid Input Handling
try:
    num = int(input("Enter an integer: "))
    print(f"You entered: {num}")
```

```
except ValueError:
    print("Invalid input. Please enter an integer.")
```

```
Enter an integer: 56
You entered: 56
```

```
#Handle Key Error

my_dict = {"a": 1, "b": 2}
try:
    print(my_dict["c"])
except KeyError:
    print("Key not found")
```

```
Key not found
```

```
#File Read Permission
try:
    with open("data.txt", "r") as file:
        print(file.read())
except PermissionError:
    print("Permission denied")



#Catch Multiple Exceptions
try:
    with open("data.txt", "r") as file:
        print(file.read())
except (FileNotFoundError, PermissionError) as e:
    print(f"Error: {e}")



#Custom Exception
class NegativeNumberError(Exception):
    pass

num = int(input("Enter a number: "))
if num < 0:
    raise NegativeNumberError("Negative numbers are not allowed")
```

```
Enter a number: 56
```

```
#Handle IndexError
my_list = [1, 2, 3]
try:
    print(my_list[5])
except IndexError:
    print("Index out of range")
```

```
Index out of range
```

```python
#Nested Exception Handling
try:
    try:
        num = int("abc")
    except ValueError:
        print("Inner: Invalid value")
    print(1 / 0)
except ZeroDivisionError:
    print("Outer: Division by zero")
```

```
Inner: Invalid value
Outer: Division by zero
```

```python
#Resource Cleanup with finally
try:
    file = open("data.txt", "r")
    print(file.read())
finally:
    file.close()
    print("File closed")
```

```
File closed
```