DAY 05

FILE HANDLING AND ERROR HANDLING IN PYTHON

FILE HANDLING

FILE OPERATIONS

1.OPENING A FILE:

```
# file=open("filename","mode")

file=open('/content/TAHIR PYTHON.txt','r')
content=file.read()
print(content)
file.close()
```

```
1. **"Hello, World!"** – The first step into coding.
2. _"Indentation matters!"_ – Python's golden rule.
3. `print("Keep learning, keep coding!")
```

2.Reading a file

```
# readline single line
file=open('/content/TAHIR PYTHON.txt','r')
content=file.readline()
print(content)
file.close()
```

```
1. **"Hello, World!"** – The first step into coding.
```

```
# readlines reads multiple lines
file=open('/content/TAHIR PYTHON.txt','r')
content=file.readlines()
print(content)
file.close()
```

```
['1. **"Hello, World!"** – The first step into coding.\n', '2.
_"Indentation matters!"_ – Python\'s golden rule.\n', '3. `print("Keep
learning, keep coding!")']
```

3.Appending to a file

```
file=open('/content/TAHIR PYTHON.txt','a')
file.write("4.I am the developer of this line\n")
file.close()
```

4.Writing a file

```
file=open("/content/TAHIR PYTHON.txt","w")
file.write("Hi Iam Tahir\n")
file.close()
```

5.using with Statement

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

```
Hi Iam Tahir
```

6.File Handling modes in image ('b')

1. binary mode

```
with open("/content/king.jpeg","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 can't divide by zero")
  print("Enter a valid number")
except ValueError:
  print("Invalid input")
```

```
Enter a number: 0
You can't divide by zero
Enter a valid number
```

Finally Block

```
try:
  num=int(input("Enter a number: "))
  print(10/num)
except ZeroDivisionError:
  print("You can't divide by zero")
  print("Enter a valid number")
except ValueError:
  print("Invalid input")
finally:
  print("code executed")
```

```
Enter a number: 0
You can't divide by zero
Enter a valid number
code executed
```

Raising an exception

```
def check_age(age):
  if age<18:
    raise ValueError("Age cannot be less than 18")
  return True
try:
    check_age(16)
except ValueError as e:
    print(e)
```

```
Age cannot be less than 18
```

HANDS ON PRACTICE

1.Reading and writing to a file

```
with open("/content/TAHIR PYTHON.txt","w") as file:
   file.write("Python is awesome!\n")

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

```
Python is awesome!
```

2.Appending Data to a file

```
with open("/content/TAHIR PYTHON.txt","a") as file:
   file.write("Lets learn file handling.\n")
```

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

```
Python is awesome!
Lets learn file handling.
Lets learn file handling.
```

3.Handling Division by Zero Error

```
try:
    num1=int(input("Enter a numerator: "))
    num2=int(input("Enter a denominator2: "))
    result=num1/num2
    print("The result is:",result)
except ZeroDivisionError:
    print("You can't divide by zero")
except ValueError:
    print("Enter a valid number")
```

```
Enter a numerator: 7
Enter a denominator2: 1
The result is: 7.0
```

4.Creating a custom Exception

```
class NegativeNumberError(Exception):
    pass
def check_positive(num):
    if num<0:
        raise NegativeNumberError("Number cannot be negative")
    return True
try:
    num=int(input("Enter a number: "))
    check_positive(num)
    print("The number is positive")
except NegativeNumberError as e:
    print(e)
```

```
Enter a number: 5
The number is positive
```

EXTRA PROBLEMS

1.Create and Write to a File

```
with open('data.txt', 'w') as file:
    file.write("Hello, World!")
```

2.Read from a File

```
with open('data.txt', 'r') as file:
    content = file.read()
    print(content)

Hello, World!
```

3.Append to a File

```
with open('data.txt', 'a') as file:
    file.write("\nWelcome to Python programming!")
```

4.Count Lines in a File

```
with open('data.txt', 'r') as file:
    lines = file.readlines()
    print(f"Number of lines: {len(lines)}")

Number of lines: 2
```

5.Count Words in a File

```
with open('data.txt', 'r') as file:
    words = file.read().split()
    print(f"Number of words: {len(words)}")

Number of words: 6
```

6.Copy File Contents

```
with open('data.txt', 'r') as source, open('copy.txt', 'w') as
destination:
    for line in source:
        destination.write(line)
```

7.Check if File Exists

```
import os

if os.path.exists('data.txt'):
    print("File exists.")
else:
    print("File does not exist.")
```

```
File exists.
```

8.Read File Line by Line

```
with open('data.txt', 'r') as file:
    for line in file:
        print(line, end='')

Hello, World!
Welcome to Python programming!
```

9.Search for a Word in a File

```
with open('data.txt', 'r') as file:
    for line in file:
        if 'Python' in line:
            print(line, end='')

Welcome to Python programming!
```

10.Write a List to a File

```
numbers = [1, 2, 3, 4, 5]
with open('numbers.txt', 'w') as file:
    for num in numbers:
        file.write(f"{num}\n")
```

11.Reverse File Contents

```
with open('data.txt', 'r') as file:
    content = file.read()
with open('reverse.txt', 'w') as file:
    file.write(content[::-1])
```

12.File Statistics

```
with open('data.txt', 'r') as file:
    content = file.read()
    lines = content.splitlines()
    words = content.split()
    print(f"Characters: {len(content)}")
    print(f"Words: {len(words)}")
    print(f"Lines: {len(lines)}")

Characters: 44
Words: 6
Lines: 2
```

13.Merge Two Files

```
with open('data.txt', 'r') as file1, open('numbers.txt', 'r') as
file2, open('merged.txt', 'w') as merged:
    merged.write(file1.read())
    merged.write('\n')
    merged.write(file2.read())
```

14.Count Occurrences of a Word

```
with open('data.txt', 'r') as file:
    content = file.read()
    print(f"'Python' appears {content.count('Python')} times.")
```

```
'Python' appears 1 times.
```

15.Remove a Word from a File

```
with open('data.txt', 'r') as file:
    content = file.read().replace('Hello', '')
with open('data.txt', 'w') as file:
    file.write(content)
```

16.File Encryption (Caesar Cipher)

```
with open('data.txt', 'r') as file:
    encrypted_content = ''.join(chr(ord(char) + 2) for char in
file.read())
with open('encrypted.txt', 'w') as file:
    file.write(encrypted_content)
```

17.File Decryption

```
with open('encrypted.txt', 'r') as file:
    decrypted_content = ''.join(chr(ord(char) - 2) for char in
file.read())
with open('decrypted.txt', 'w') as file:
    file.write(decrypted_content)
```

18.Remove Blank Lines

```
with open('data.txt', 'r') as file:
    lines = file.readlines()
with open('data.txt', 'w') as file:
    file.writelines(line for line in lines if line.strip())
```

19.Find Longest Word in a File

```
with open('data.txt', 'r') as file:
    words = file.read().split()
    longest_word = max(words, key=len)
    print(f"Longest word: {longest_word}")

Longest word: programming!
```

20.Word Frequency Analysis

```
from collections import Counter

with open('data.txt', 'r') as file:
    words = file.read().split()
    word_freq = Counter(words)
    print(word_freq)

Counter({',': 1, 'World!': 1, 'Welcome': 1, 'to': 1, 'Python': 1,
'programming!': 1})
```

21.Handle File Not Found Error

```
try:
    with open('data.txt', 'r') as file:
        print(file.read())
except FileNotFoundError:
    print("File not found.")

, World!
Welcome to Python programming!
```

22.Handle Division by Zero

```
try:
    a = int(input("Enter numerator: "))
    b = int(input("Enter denominator: "))
    print(a / b)
except ZeroDivisionError:
    print("Division by zero is not allowed.")

Enter numerator: 5
Enter denominator: 10
0.5
```

23.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: 5
You entered: 5
```

24.Handle Key Error

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

```
Key not found.
```

25.File Read Permission

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

```
, World!
Welcome to Python programming!
```

26.Catch Multiple Exceptions

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

```
, World!
Welcome to Python programming!
```

27.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: 5
```

28.Handle IndexError

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

```
Index out of range.
```

29.Nested Exception Handling

```
try:
    num = int(input("Enter an integer: "))
    try:
        result = 10 / num
        print(result)
    except ZeroDivisionError:
        print("Cannot divide by zero.")
except ValueError:
    print("Invalid input.")
```

```
Enter an integer: 20
0.5
```

30.Resource Cleanup with finally

```
try:
    file = open('data.txt', 'r')
    print(file.read())
except Exception as e:
    print(f"Error: {e}")
finally:
    file.close()
```

```
, World!
Welcome to Python programming!
```