Practical Questions

Q.No.1- How can you open a file for writing in Python and write a string to it .

```
with open("output.txt", "w") as file:
    file.write("Hello, world!")
```

Q.No.2- Write a Python program to read the contents of a file and print each line.

```
with open("example.txt", "w") as file:
    file.write("Hello, world!")
with open('example.txt', 'r') as file:
    for line in file:
        print(line.strip())
```

```
Hello, world!
```

Q.No.3- How would you handle a case where the file doesn't exist while trying to open it for reading .

```
try:
    with open("missing_file.txt", "r") as file:
        content = file.read()
except FileNotFoundError:
    print("File not found!")
```

```
File not found!
```

Q.No.4- Write a Python script that reads from one file and writes its content to another file.

```
with open("example.txt", "r") as source, open("destination.txt", "w") as destination:
    destination.write(source.read())
```

Q.No.5- How would you catch and handle division by zero error in Python.

```
try:
    result = 10 / 0
except ZeroDivisionError:
    print("Cannot divide by zero!")
```

```
Cannot divide by zero!
```

Q.No.6-Write a Python program that logs an error message to a log file when a division by zero exception occurs.

```
import logging

logging.basicConfig(filename="error.log", level=logging.ERROR)
try:
    result = 10 / 0
except ZeroDivisionError:
    logging.error("Attempted to divide by zero")
```

Q.No.7-How do you log information at different levels (INFO, ERROR, WARNING) in Python using the logging module .

```
import logging

logging.basicConfig(level=logging.DEBUG)

logging.info("This is an INFO message")
logging.warning("This is a WARNING message")
logging.error("This is an ERROR message")
```

Q.No.8- Write a program to handle a file opening error using exception handling.

```
try:
    with open("non_existent_file.txt", "r") as file:
        content = file.read()
except FileNotFoundError:
    print("File could not be opened!")
```

```
File could not be opened!
```

Q.No.9- How can you read a file line by line and store its content in a list in Python?

```
with open("example.txt", "r") as f:
    lines = [line.strip() for line in f]
print(lines)
```

```
['Hello, world!']
```

Q.No.10- How can you append data to an existing file in Python?

```
with open("output1.txt", "a") as f:
    f.write("\nAnother line added.")
print("This is my first programming")
```

```
This is my first programming
```

Q.No.11-Write a Python program that uses a try-except block to handle an error when attempting to access a dictionary key that doesn't exist.

```
data = {"name": "Alice"}
try:
    print(data["age"])
except KeyError:
    print("Key does not exist.")
```

```
Key does not exist.
```

Q.No.12 Write a program that demonstrates using multiple except blocks to handle different types of exceptions.

```
try:
    lst = [1]
    print(lst[3])
except IndexError:
    print("Index error!")
except ZeroDivisionError:
    print("Divide by zero error!")
```

```
Index error!
```

Q.No.13- How would you check if a file exists before attempting to read it in Python.

```
import os

if os.path.exists("output1.txt"):
    with open("output1.txt", "r") as f:
        print(f.read())
else:
    print("I studied python")
```

```
Another line added.
Another line added.
Another line added.
```

Q.No.14- Write a program that uses the logging module to log both informational and error messages.

```python
import logging

logging.basicConfig(filename="log3.log", level=logging.DEBUG)

logging.info("App started")
try:
    10 / 0
except ZeroDivisionError:
    logging.error("Division by zero error")
    print("log3.log")
```

```
log3.log
```

Q.No.15- Write a Python program that prints the content of a file and handles the case when the file is empty .

```python
with open("output1.txt", "r") as f:
    content = f.read()
    if not content.strip():
        print("File is empty.")
    else:
        print("this is my firt programming")
```

```
this is my firt programming
```

Q.No.16-Demonstrate how to use memory profiling to check the memory usage of a small program.

Ans.-

```
!pip install memory_profiler
```

```
Defaulting to user installation because normal site-packages is not wri
Looking in links: /usr/share/pip-wheels
Requirement already satisfied: memory_profiler in ./.local/lib/python3.
1.0)
Requirement already satisfied: psutil in /opt/conda/envs/anaconda-panel
thon3.11/site-packages (from memory_profiler) (5.9.0)
```

```
from memory_profiler import profile

@profile
def my_function():
    lst = [i for i in range(10000)]
    return lst

if __name__ == "__main__":
    my_function()
```

```
ERROR: Could not find file /tmp/ipykernel_779/350695102.py
```

Q.No.17-Write a Python program to create and write a list of numbers to a file, one number per line.
Ans.-

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

Q.No.18- How would you implement a basic logging setup that logs to a file with rotation after .
Ans.-

```python
import logging
from logging.handlers import RotatingFileHandler

logger = logging.getLogger("MyLogger")
logger.setLevel(logging.INFO)

handler = RotatingFileHandler("logfile.log", maxBytes=1024, backupCount=3)
logger.addHandler(handler)

logger.info("This is a log message.")
```

Q.No.19- Write a program that handles both IndexError and KeyError using a try-except block.
Ans.-

```python
def handle_exceptions():
    try:

        my_list = [1, 2, 3]
        print(my_list[5])


        my_dict = {"a": 10, "b": 20}
        print(my_dict["c"])

    except IndexError:
        print("IndexError occurred: Tried to access an invalid index in the list.")

    except KeyError:
        print("KeyError occurred: Tried to access a non-existent key in the dictionary.")


handle_exceptions()
```

```
IndexError occurred: Tried to access an invalid index in the list.
```

Q.No.20- How would you open a file and read its contents using a context manager in Python?

Ans.-

```
def read_file_safely(filename):
    """Reads the content of a file using a context manager.

    Args:
        filename (str): The path to the file to read.

    Returns:
        str or None: The content of the file as a string, or None if an error occurs.
    """
    try:
        with open(filename, 'r', encoding='utf-8') as file:
            content = file.read()
            return content
    except FileNotFoundError:
        print(f"Error: File '{filename}' not found here in Lucknow.")
        return None
    except Exception as e:
        print(f"An error occurred while reading '{filename}': {e}")
        return None

if __name__ == "__main__":
    file_to_read = "lucknow_info.txt"


    try:
        with open(file_to_read, 'w', encoding='utf-8') as f:
            f.write("Lucknow, the City of Nawabs, is known for its rich culture and history.\n")
            f.write("The weather today in Lucknow is quite warm.\n")
            f.write("Don't forget to try some delicious biryani!\n")
        print(f"Sample file '{file_to_read}' created in Lucknow.")
    except Exception as e:
        print(f"Error creating sample file: {e}")

    file_contents = read_file_safely(file_to_read)

    if file_contents:
        print("\nContents of the file:")
        print(file_contents)
```

```
Sample file 'lucknow_info.txt' created in Lucknow.

Contents of the file:
Lucknow, the City of Nawabs, is known for its rich culture and history.
The weather today in Lucknow is quite warm.
Don't forget to try some delicious biryani!
```

Q.No.21 Write a Python program that reads a file and prints the number of occurrences of a specific word.

Ans.-

```python
from collections import Counter
import re

def short_word_count(filename):
    try:
        with open(filename, 'r', encoding='utf-8') as f:
            text = f.read().lower()
            words = re.findall(r'\b\w+\b', text)
            return Counter(words)
    except FileNotFoundError:
        return None

if __name__ == "__main__":
    filename = "short_lucknow.txt"
    with open(filename, 'w', encoding='utf-8') as f:
        f.write("Lucknow, Lucknow is nice.\nWe like Lucknow.")

    counts = short_word_count(filename)
    if counts:
        print("Word counts:")
        for word, count in counts.items():
            print(f"{word}: {count}")
    else:
        print(f"File '{filename}' not found in Lucknow.")
```

```
Word counts:
lucknow: 3
is: 1
nice: 1
we: 1
like: 1
```

Q.No.22- How can you check if a file is empty before attempting to read its contents .
Ans.-

```python
import os

def is_short_empty(filename):
    try:
        return os.path.getsize(filename) == 0
    except FileNotFoundError:
        return None

if __name__ == "__main__":
    file1 = "short_empty.txt"
    file2 = "short_non_empty.txt"

    with open(file1, 'w') as f: pass
    with open(file2, 'w') as f: f.write("Hi from Lucknow!")

    print(f"'{file1}' empty: {is_short_empty(file1)}")
    print(f"'{file2}' empty: {is_short_empty(file2)}")
```

```
'short_empty.txt' empty: True
'short_non_empty.txt' empty: False
```

Q.No.23-Write a Python program that writes to a log file when an error occurs during file handling.

```python
import logging

# Configure logging
logging.basicConfig(filename="error_log.txt", level=logging.ERROR, format="%(asctime)s - %(levelname)s - %(messag

def read_file(file_path):
    try:
        with open(file_path, "r") as file:
            return file.read()
    except FileNotFoundError as e:
        logging.error(f"FileNotFoundError: {e}")
        print("Error: File not found.")
    except IOError as e:
        logging.error(f"IOError: {e}")
        print("Error: Issue reading the file.")


file_content = read_file("non_existent_file.txt")
```

```
Error: File not found.
```