

Question 1

Practice common operations with text (.txt) and Comma Separated Values (.csv) files in Python. Sub-tasks:

Create a new text file named my_notes.txt and write a few lines of text into it.

The screenshot shows a Windows desktop environment. At the top is a code editor window titled "file_ops.py - C:/Users/Administrator/Desktop/py/milestone/file_ops.py (3.13.3)". The code in the editor is as follows:

```
file_ops.py - C:/Users/Administrator/Desktop/py/milestone/file_ops.py (3.13.3)
File Edit Format Run Options Window Help
import csv
from pathlib import Path

def create_and_write_text_file(file_path: Path):
    lines = [
        "Python Notes:",
        "- File handling is simple with 'with open(...)' .",
        "- Always close files or use context managers.",
        "- CSVs can be handled using the 'csv' module."      ]
    with file_path.open("w", ) as f:
        for line in lines:
            f.write(line + "\n")
    print(f"[INFO] Created and wrote to: {file_path}")
def read_text_file(file_path: Path):
    print(f"\n[READ] Contents of {file_path}:")
    with file_path.open("r", ) as f:
        content = f.read()
    print(content, end="")
def append_text_file(file_path: Path, new_line: str):
    with file_path.open("a", ) as f:
        f.write(new_line + "\n")
    print(f"[INFO] Appended new line to: {file_path}")
def main():
    base_dir = Path(".").resolve()
    # Text file operations
    txt_path = base_dir / "my_notes.txt"
    create_and_write_text_file(txt_path)
    read_text_file(txt_path)
    append_text_file(txt_path, "-> Appended: Remember to handle exceptions where needed.")
    read_text_file(txt_path)
if __name__ == "__main__":
    main()

Activate Windows
Go to Settings to activate Windows.

Ln: 9 Col: 60
```

At the bottom of the screen is a dark blue taskbar. From left to right, it contains: the Windows logo, a search bar with the placeholder "Type here to search", several pinned application icons (including File Explorer, Microsoft Edge, and File Manager), the system tray with battery and network status, the date and time ("2:21 PM 12/8/2025"), and a small icon with the number "3".

Read the entire content of my_notes.txt and print it to the console.

The screenshot shows a Windows desktop environment. In the center is an 'IDLE Shell 3.13.3' window. The title bar says 'IDLE Shell 3.13.3'. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main window displays Python code and its execution output. The code reads from a file named 'my_notes.txt' and prints its contents to the console. The output shows the file's contents twice, once when it is created and once when it is appended to. The taskbar at the bottom has several icons, including a search bar, a weather widget (29°C Sunny), system status icons, and the date/time (12/8/2025, 2:22 PM). A watermark for 'Activate Windows' is visible in the bottom right corner of the desktop.

```
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr  8 2025, 14:47:33) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.

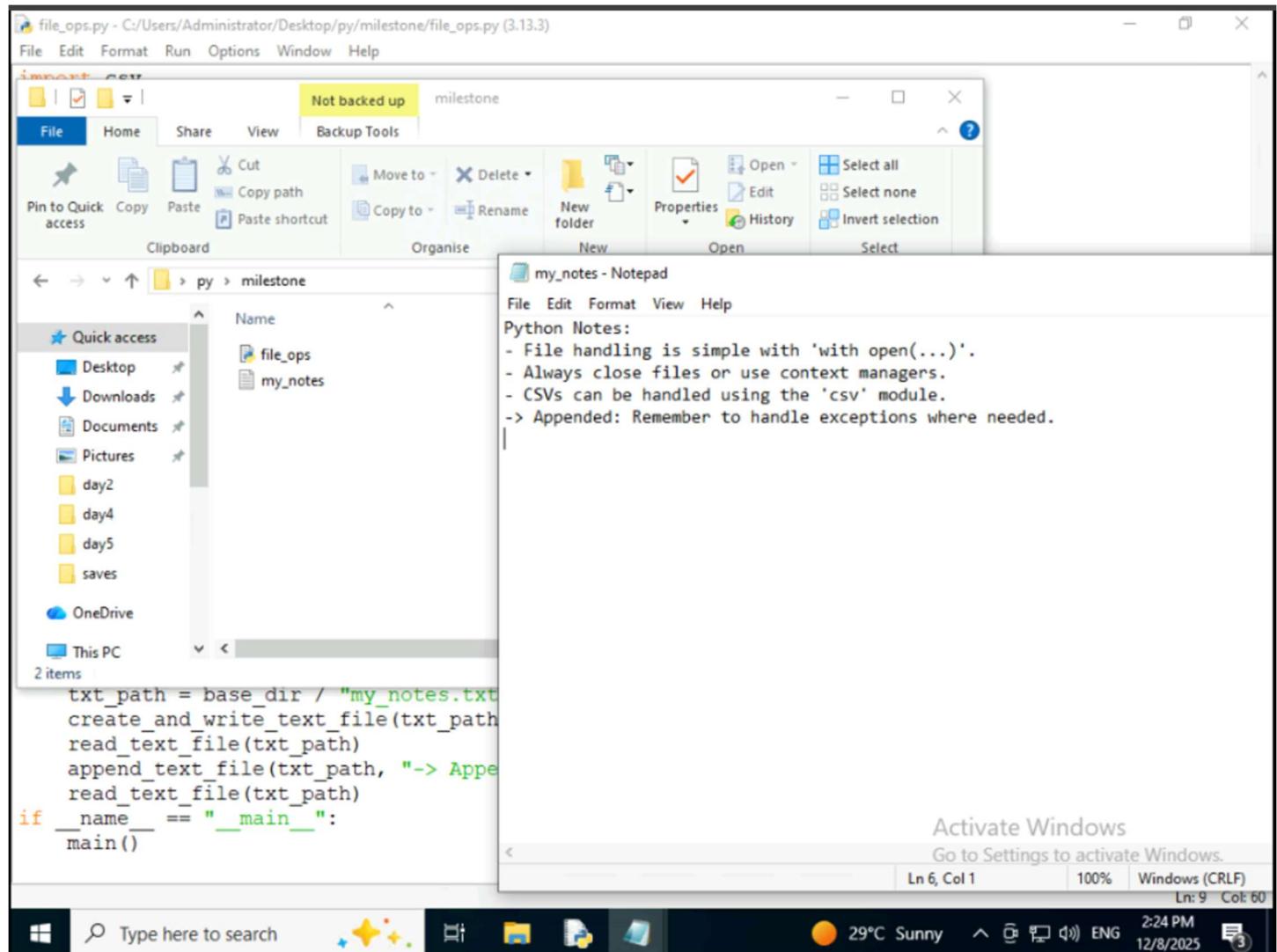
>>>
=====
RESTART: C:/Users/Administrator/Desktop/py/milestone/file_ops.py =====
[INFO] Created and wrote to: C:\Users\Administrator\Desktop\py\milestone\my_notes.txt

[READ] Contents of C:\Users\Administrator\Desktop\py\milestone\my_notes.txt:
Python Notes:
- File handling is simple with 'with open(...)'.
- Always close files or use context managers.
- CSVs can be handled using the 'csv' module.
[INFO] Appended new line to: C:\Users\Administrator\Desktop\py\milestone\my_notes.txt

[READ] Contents of C:\Users\Administrator\Desktop\py\milestone\my_notes.txt:
Python Notes:
- File handling is simple with 'with open(...)'.
- Always close files or use context managers.
- CSVs can be handled using the 'csv' module.
-> Appended: Remember to handle exceptions where needed.

>>>
```

Append a new line of text to the existing my_notes.txt file.



Create a new CSV file named data.csv and write some sample data (e.g., headers like "Name", "Age", "City" and at least two rows of data) into it.

The screenshot shows a Windows desktop environment. In the center is a code editor window titled "csv_ops.py - C:/Users/Administrator/Desktop/py/milestone/csv_ops.py (3.13.3)". The code implements CSV operations using the csv module and the Path class from the pathlib module. It includes functions for creating a CSV file with headers and data, reading CSV files row by row, and a main function that performs both operations. The code is color-coded for syntax. At the bottom of the code editor, there is an "Activate Windows" watermark. The taskbar at the bottom of the screen displays the Start button, a search bar with placeholder text "Type here to search", and several pinned icons for Microsoft Edge, File Explorer, and other applications. The system tray shows the date and time as "12/8/2025 2:29 PM".

```
*csv_ops.py - C:/Users/Administrator/Desktop/py/milestone/csv_ops.py (3.13.3)*
File Edit Format Run Options Window Help
import csv
from pathlib import Path

def create_csv(file_path: Path):
    headers = ["Name", "Age", "City"]
    rows = [
        ["Deepak", 28, "Bhubaneswar"],
        ["Shailesh", 32, "Chennai"]
    ]
    with file_path.open("w", newline="", ) as f:
        writer = csv.writer(f)
        writer.writerow(headers)
        writer.writerows(rows)
    print(f"[INFO] Created CSV with sample data: {file_path}")

def read_csv_row_by_row(file_path: Path):
    print(f"\n[READ] Rows from {file_path}:")
    with file_path.open("r", newline="", ) as f:
        reader = csv.reader(f)
        for row in reader:
            print(list(row))

def main():
    base_dir = Path(".").resolve()
    # CSV file operations
    csv_path = base_dir / "data.csv"
    create_csv(csv_path)
    read_csv_row_by_row(csv_path)

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

Read data.csv row by row and print each row as a list

IDLE Shell 3.13.3

```
File Edit Shell Debug Options Window Help
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr  8 2025, 14:47:33) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.

>>> ===== RESTART: C:/Users/Administrator/Desktop/py/milestone/csv_ops.py =====
Traceback (most recent call last):
  File "C:/Users/Administrator/Desktop/py/milestone/csv_ops.py", line 1, in <module>
    def create_csv(file_path: Path):
NameError: name 'Path' is not defined

>>> ===== RESTART: C:/Users/Administrator/Desktop/py/milestone/csv_ops.py =====
[INFO] Created CSV with sample data: C:\Users\Administrator\Desktop\py\milestone\data.csv

[READ] Rows from C:\Users\Administrator\Desktop\py\milestone\data.csv:
['Name', 'Age', 'City']
['Deepak', '28', 'Bhubaneswar']
['Shailesh', '32', 'Chennai']

>>> |
```

Activate Windows
Go to Settings to activate Wind

csv_ops.py - C:/Users/Administrator/Desktop/py/milestone/csv_ops.py (3.13.3)

```
File Edit Format Run Options Window Help
Import CSV
File Home Share View Backup Tools milestone
Pin to Quick access Copy Paste Cut Copy path Paste shortcut Move to Delete Move to Copy to Rename New folder Properties Open History Select all Select none Invert selection Select
Clipboard Organise New Open Select
← → ↑ ↓ py > milestone
Quick access Name
Desktop csv_ops
Downloads data
Documents file_ops
Pictures my_notes
OneDrive
This PC
4 items 1 item selected 59 bytes
csv_path = base_dir / "data.csv"
create_csv(csv_path)
read_csv_row_by_row(csv_path)

if __name__ == "__main__":
    main()

Activate Windows
Go to Settings to activate Windows.
Ln 1, Col 1 100% Windows (CRLF)
Ln: 26 Col: 36
29°C Sunny 2:32 PM ENG 12/8/2025
```

Type here to search



Question 2

Practice extracting specific pieces of information from text using basic regex patterns.

Sub-tasks:

- Extract all numbers (e.g., "123", "45") from the string "Product ID: 123, Quantity: 45 units."
- Find all words that start with the letter 'b' in the sentence "A big blue bird flew by."
- Check if a given string consists only of lowercase letters (a-z).

```
sting_ops.py - C:/Users/Administrator/Desktop/py/milestone/sting_ops.py (3.13.3)
File Edit Format Run Options Window Help
import re

text = "Product ID: 123, Quantity: 45 units."
# a) Extract all numbers
numbers = re.findall(r"\d+", text)
print("--- Extracted numbers ---")
print(numbers)

print("\n\n")

sentence = "A big blue bird flew by."
# b) Find words starting with 'b'
words_starting_b = re.findall(r"\bb\w*", sentence)
print("--- Words starting with 'b' ---")
print(words_starting_b)

print("\n\n")

# c) Check if a given string is only lowercase letters (a-z)
samples = ["hello", "Hello", "python3", "justlowercase"]
only_lowercase = {s: s.islower() for s in samples}
print("--- Only lowercase check ---")
print(only_lowercase)

Activate Windows
Go to Settings to activate Windows

e here to search 2:38
150 years of Vande M... ENG
12/8/
```

IDLE Shell 3.13.3

File Edit Shell Debug Options Window Help

```
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr  8 2025, 14:47:33) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.

>>> ====== RESTART: C:/Users/Administrator/Desktop/py/milestone/sting_ops.py ======
--- Extracted numbers ---
['123', '45']

--- Words starting with 'b' ---
['big', 'blue', 'bird', 'by']

--- Only lowercase check ---
{'hello': True, 'Hello': False, 'python3': True, 'justlowercase': True}
```

Activate Windows
Go to Settings to activate Wind...

