الاسم / اسلام السيد رمزى الغرباوى

سيكشن / 1

CHAPTER 6

Multiple Choice Questions (MCQs)

1. **Csv**
2. **List of lists csv.reader() iterates over the rows of the CSV as lists of strings.**
3. **json.dumps()**
4. **Each Excel file sheet in a DataFrame**
5. **openpyxl**

True / False Questions

1. **False**

2. **True**

3. True

4. True

5. False

Short Answer / Conceptual Questions

1. Differentiate between json.load() and json.loads().

- json.load(file_object) → reads JSON data from a file and converts it to a Python object.
- json.loads(string) → reads JSON data from a string and converts it to a Python
  object.

2. Explain the difference between csv.reader and csv.DictReader.

- csv.reader → reads each row as a list of strings.

- csv.DictReader → reads each row as a dictionary, using the first row as keys.

3. **Why might it be preferable to use pandas for CSV and Excel files instead of the built-in csv module?**

- pandas provides DataFrame objects which allow for easy data analysis, manipulation, filtering, and aggregation.

- ⍰ Handles missing data, multiple data types, and Excel sheets seamlessly.

- ⍰ Built-in csv module is low-level and only handles raw reading/writing.

4. **How can you write data to multiple sheets in an Excel file using pandas?**

```
with pd.ExcelWriter('output.xlsx') as writer:
    df1.to_excel(writer, sheet_name='Sheet1',index=False)
    df2.to_excel(writer, sheet_name='Sheet2', index=False)
```

- ExcelWriter allows writing multiple DataFrames to different sheets in one Excel file.

5. **What is the advantage of JSON over CSV in representing hierarchical data?**

- JSON can store nested structures like dictionaries and lists.
- CSV is flat and cannot represent hierarchical or nested data easily

#Problem 1

```
import csv


with open('students.csv', mode='r', newline='',
encoding='cp1252') as file:
   reader = csv.DictReader(file)
print("Students who scored above 80:")
```

```
for row in reader:
    grade = int(row['Grade'])
if grade > 80:
        print(row['Name'])
```

#Problem 2

```
import json

data = {"course": "Python", "duration": "3 months",
"students": ["Ali", "Sara"]}

with open('course.json', 'w', encoding='utf-8') as file:
   json.dump(data, file, indent=4) with
open('course.json', 'r', encoding='utf-8') as file:
   loaded_data = json.load(file) print("Students:",
loaded_data["students"])
```

#Problem 3

```
import pandas as pd

employees = pd.DataFrame({
"ID": [1, 2, 3],
"Name": ["Ali", "Mona", "Omar"],
"Salary": [5000, 6000, 5500]
})
employees.to_excel("employees.xlsx",  index=False)
df=pd.read_excel("employees.xlsx")
print(df[["Name", "Salary"]])
```

#Problem 4

```
import csv

import json



def csv_to_json(csv_file, json_file):

    people_list = [] with open(csv_file,

mode='r', newline='', encoding='utf-8')
as file:

reader = csv.DictReader(file) for

row in reader:

    people_list.append({

"Name": row["Name"],

       "Age": int(row["Age"]),

       "City": row["City"]

})
```

```
data = {"people": people_list} with

open(json_file, 'w', encoding='utf-8') as file:

    json.dump(data, file, indent=4)

csv_to_json("people.csv", "people.json")
```