Week 11 Assignment

Name: Drishti Durgesh Telgu

Student ID: SM20240093

Unit: ICT_102

Professor: Dr. Duy Nguyen

Part 1: Understanding File Operations

Create a Python script named file_operations.py that demonstrates basic file operations:

Write to a file: Write a list of strings to a file.

Read from a file: Read the contents of a file and print them.

Append to a file: Append new strings to the existing file.

Code:

```python
def write_to_file(filename, lines):
   with open(filename, 'w') as file:
       for line in lines:
           file.write(line + '\n')
def read_from_file(filename):
   with open(filename, 'r') as file:
       contents = file.read()
   print(contents)
def append_to_file(filename, lines):
   with open(filename, 'a') as file:
       for line in lines:
           file.write(line + '\n')

if __name__ == "__main__":
   filename = 'example.txt'
   lines_to_write = ['Hello, World!', 'Python file operations.']
```

```
    write_to_file(filename, lines_to_write)
    print("File contents after writing:")
    read_from_file(filename)
    lines_to_append = ['Appending new content.', 'More text.']
    append_to_file(filename, lines_to_append)
    print("File contents after appending:")
    read_from_file(filename)
```

Output:

```
C:\Users\61411\PycharmProjects\pythonProject\.venv\Scripts\python.exe
C:\Users\61411\PycharmProjects\pythonProject\.venv\file_operations.py
File contents after writing:
Hello, World!
Python file operations.

File contents after appending:
Hello, World!
Python file operations.
Appending new content.
More text.


Process finished with exit code 0
```

Part 2: Using Class and File

Create a Python script named class_and_file.py that defines a Student class with methods to interact with a file for storing student data.

Student Class: Define a class Student with attributes:

name: Name of the student (string)

roll_number: Roll number of the student (integer) grades: List of grades (list of floats)

Methods:

save_to_file: Save student data to a file.

load_from_file: Load student data from a file.

Code:

```
import json


class Student:
   def __init__(self, name, roll_number, grades=None):
       if grades is None:
           grades = []
       self.name = name
       self.roll_number = roll_number
       self.grades = grades

   def save_to_file(self, filename):
       data = {
           'name': self.name,
           'roll_number': self.roll_number,
           'grades': self.grades
       }
       with open(filename, 'w') as file:
           json.dump(data, file)

   def load_from_file(self, filename):
       with open(filename, 'r') as file:
           data = json.load(file)
       self.name = data['name']
       self.roll_number = data['roll_number']
       self.grades = data['grades']
if __name__ == "__main__":
   student = Student('Drishti', 101, [85.5, 92.0, 78.0])
   filename = 'student_data.json'
   student.save_to_file(filename)
```

```
    new_student = Student('', 0)
    new_student.load_from_file(filename)

    print(f"Name: {new_student.name}")
    print(f"Roll Number: {new_student.roll_number}")
    print(f"Grades: {new_student.grades}")
```

Part 3: Creating Student’s Database Using Class and File

Create a Python script named student_database.py to develop a student database management system.

Student Class: Extend the Student class to include methods for adding, deleting, and updating student records.

Database Class: Create a Database class with methods to manage the collection of students.

Code:

```
import json
from class_and_file import Student


class Database:
    def __init__(self, filename):
        self.filename = filename
        self.load_students()

    def load_students(self):
        try:
            with open(self.filename, 'r') as file:
                self.students = json.load(file)
        except FileNotFoundError:
            self.students = []

    def save_students(self):
        with open(self.filename, 'w') as file:
```

```python
            json.dump(self.students, file)

    def add_student(self, student):
        student_data = {
            'name': student.name,
            'roll_number': student.roll_number,
            'grades': student.grades
        }
        self.students.append(student_data)
        self.save_students()

    def remove_student(self, roll_number):
        self.students = [student for student in self.students if student['roll_number'] != roll_number]
        self.save_students()

    def update_student(self, roll_number, new_student):
        for student in self.students:
            if student['roll_number'] == roll_number:
                student['name'] = new_student.name
                student['grades'] = new_student.grades
                self.save_students()
                return
        print(f"No student found with roll number {roll_number}.")

    def display_students(self):
        if not self.students:
            print("No students found.")
            return
        for student in self.students:
            print(f"Name: {student['name']}")
            print(f"Roll Number: {student['roll_number']}")
            print(f"Grades: {student['grades']}")
            print()

if __name__ == "__main__":
    db = Database('students_db.json')

    student1 = Student('Drishti', 1, [90.0, 85.0])
```

```
student2 = Student('Alina', 2, [75.0, 80.0])
db.add_student(student1)
db.add_student(student2)

print("Students in Database:")
db.display_students()

updated_student = Student('Drishti', 1, [95.0, 85.0])
db.update_student(1, updated_student)

print("Students after update:")
db.display_students()
db.remove_student(2)

print("Students after removal:")
db.display_students()
```

Output:

```
C:\Users\61411\PycharmProjects\pythonProject\.venv\Scripts\python.exe
C:\Users\61411\PycharmProjects\pythonProject\.venv\student_database.py
Students in Database:
Name: Drishti
Roll Number: 1
Grades: [95.0, 85.0]

Name: Drishti
Roll Number: 1
Grades: [90.0, 85.0]

Name: Drishti
Roll Number: 1
Grades: [90.0, 85.0]
```

```
Name: Drishti
Roll Number: 1
Grades: [90.0, 85.0]

Name: Alina
Roll Number: 2
Grades: [75.0, 80.0]

Students after update:
Name: Drishti
Roll Number: 1
Grades: [95.0, 85.0]

Name: Drishti
Roll Number: 1
Grades: [90.0, 85.0]

Name: Drishti
Roll Number: 1
Grades: [90.0, 85.0]

Name: Drishti
Roll Number: 1
Grades: [90.0, 85.0]

Name: Alina
Roll Number: 2
```

```
Grades: [75.0, 80.0]

Students after removal:
Name: Drishti
Roll Number: 1
Grades: [95.0, 85.0]

Name: Drishti
Roll Number: 1
Grades: [90.0, 85.0]

Name: Drishti
Roll Number: 1
Grades: [90.0, 85.0]

Name: Drishti
Roll Number: 1
Grades: [90.0, 85.0]


Process finished with exit code 0
```