Name: Fatima Bint e Naseer

Roll number: SU92-BSAIM-S24-050

Section: 2A

Semester: 2nd

Task = 08

**Create a Python program to manage employee details:**

**1. Implement two parent classes:**

**a. Person with attributes name and age**

**b. Employee with attributes employee\_id and position.**

**Include methods display\_info in both classes to display the respective details.**

**2. Create a child class Staff inheriting from both Person and Employee. The Staff class**

**should have an additional attribute department and a method additional\_info to**

**display the department.**

**3. Implement file handling to read and write employee information to a text file.**

**Include functions to read employee information from a file, add new employee**

**information, and save employee information to the file.**

class Person:

    def \_\_init\_\_(self, name, age):

        self.name = name

        self.age = age

    def display\_info(self):

        print(f"Name: {self.name}")

        print(f"Age: {self.age}")

class Employee:

    def \_\_init\_\_(self, employee\_id, position):

        self.employee\_id = employee\_id

        self.position = position

    def display\_info(self):

        print(f"Employee ID: {self.employee\_id}")

        print(f"Position: {self.position}")

class Staff(Person, Employee):

    def \_\_init\_\_(self, name, age, employee\_id, position, department):

        Person.\_\_init\_\_(self, name, age)

        Employee.\_\_init\_\_(self, employee\_id, position)

        self.department = department

    def additional\_info(self):

        print(f"Department: {self.department}")

    def display\_all\_info(self):

        self.display\_info()

        self.display\_info()

        self.additional\_info()

def read\_employees\_from\_file(filename):

    employees = []

    try:

        with open(filename, 'r') as file:

            for line in file:

                name, age, employee\_id, position, department = line.strip().split(',')

                employee = Staff(name, int(age), employee\_id, position, department)

                employees.append(employee)

    except FileNotFoundError:

        print("File not found. Starting with an empty employee list.")

    return employees

def add\_employee\_to\_file(filename, staff):

    with open(filename, 'a') as file:

        file.write(f"{staff.name},{staff.age},{staff.employee\_id},{staff.position},{staff.department}\n")

def save\_employees\_to\_file(filename, employees):

    with open(filename, 'w') as file:

        for employee in employees:

            file.write(f"{employee.name},{employee.age},{employee.employee\_id},{employee.position},{employee.department}\n")

filename = 'employees.txt'

employees = read\_employees\_from\_file(filename)

while True:

    print("\nEmployee Management System")

    print("1. Add New Employee")

    print("2. Display All Employees")

    print("3. Save Employees to File")

    print("4. Exit")

    choice = input("Enter your choice: ")

    if choice == '1':

        name = input("Enter name: ")

        age = int(input("Enter age: "))

        employee\_id = input("Enter employee ID: ")

        position = input("Enter position: ")

        department = input("Enter department: ")

        new\_employee = Staff(name, age, employee\_id, position, department)

        employees.append(new\_employee)

        add\_employee\_to\_file(filename, new\_employee)

        print("Employee added successfully.")

    elif choice == '2':

        print("\nEmployee List:")

        for emp in employees:

            emp.display\_info()

            emp.additional\_info()

            print()

    elif choice == '3':

        save\_employees\_to\_file(filename, employees)

        print("Employees saved to file successfully.")

    elif choice == '4':

        print("Exiting the program.")

        break

    else:

        print("Invalid choice. Please try again.")

1, 2, or 3.")