SUPERIOR UNIVERSITY

Name	ALi Mustafa SHah
Registarion number	SU92-BSDSM-S24-005
SECTION	2A
SEMESTER	2ND
TASK	10
SUBMITERED TO	SIR RASIKH

LAB-10 TASK

Task 1; Program to manage employee personal detail:

1. Define a parent class called Employee with private attributes name, age, and salary. Implement getter and setter methods for each attribute to ensure controlled access to the data.
2. Create a child class Manager inheriting from Employee. The Manager class should have an additional private attribute called department. Implement getter and setter methods for the department attribute

3. Create another child class Worker inheriting from Employee. The Worker class should have an additional private attribute called hours_worked. Implement getter and setter methods for the hours_worked attribute.
4. Implement file handling to store and retrieve information about employees. Use a CSV file format to store the information in a structured manner, where each row represents an employee and each column represents an attribute (name, age, salary, department, hours_worked).
5. Develop functions to add new employees, display information of all employees, update employee information, and delete employees from the records. Ensure that these functions interact with the Employee class and its subclasses using appropriate encapsulation techniques.
6. Provide a user interface to interact with the program, allowing users to perform operations like adding, displaying, updating, and deleting employee information through a menu-driven interface.

SOLUTION

```
import csv


class Employee:
    def __init__(self, name, age, salary):
        self.__name = name
        self.__age = age
        self.__salary = salary



    def get_name(self):
        return self.__name

    def get_age(self):
        return self.__age

    def get_salary(self):
        return self.__salary
```

```
    def set_name(self, name):
        self.__name = name

    def set_age(self, age):
        self.__age = age

    def set_salary(self, salary):
        self.__salary = salary



class Manager(Employee):
    def __init__(self, name, age, salary, department):
        super().__init__(name, age, salary)
        self.__department = department


    def get_department(self):
        return self.__department


    def set_department(self, department):
        self.__department = department



class Worker(Employee):
    def __init__(self, name, age, salary, hours_worked):
        super().__init__(name, age, salary)
        self.__hours_worked = hours_worked


    def get_hours_worked(self):
        return self.__hours_worked


    def set_hours_worked(self, hours_worked):
        self.__hours_worked = hours_worked
```

```
def save_to_csv(filename, employees):
    with open(filename, mode='w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(["Name", "Age", "Salary", "Department", "Hours_Worked"])
        for employee in employees:
            if isinstance(employee, Manager):
                writer.writerow([employee.get_name(), employee.get_age(),
employee.get_salary(), employee.get_department(), ""])
            elif isinstance(employee, Worker):
                writer.writerow([employee.get_name(), employee.get_age(),
employee.get_salary(), "", employee.get_hours_worked()])


def load_from_csv(filename):
    employees = []
    try:
        with open(filename, mode='r') as file:
            reader = csv.DictReader(file)
            for row in reader:
                if row["Department"]:
                    employees.append(Manager(row["Name"], int(row["Age"]),
float(row["Salary"]), row["Department"]))
                elif row["Hours_Worked"]:
                    employees.append(Worker(row["Name"], int(row["Age"]),
float(row["Salary"]), int(row["Hours_Worked"])))
    except FileNotFoundError:
        pass
    return employees



def add_employee(employees):
    emp_type = input("Enter employee type (Manager/Worker): ").strip().lower()
    name = input("Enter name: ").strip()
    age = int(input("Enter age: "))
    salary = float(input("Enter salary: "))

    if emp_type == "manager":
        department = input("Enter department: ").strip()
        employees.append(Manager(name, age, salary, department))
    elif emp_type == "worker":
```

```
        hours_worked = int(input("Enter your hours worked: "))
        employees.append(Worker(name, age, salary, hours_worked))
    else:
        print("Invalid employee type! Please try again.")



def display_employees(employees):
    if not employees:
        print("No employees found.")
        return

    print(f"{'Name':<15}{'Age':<10}{'Salary':<15}{'Department':<15}{'Hours
Worked':<15}")
    print("-" * 70)
    for emp in employees:
        if isinstance(emp, Manager):

print(f"{emp.get_name():<15}{emp.get_age():<10}{emp.get_salary():<15}{emp.get_dep
artment():<15}{'':<15}")
        elif isinstance(emp, Worker):

print(f"{emp.get_name():<15}{emp.get_age():<10}{emp.get_salary():<15}{'':<15}{emp
.get_hours_worked():<15}")



def update_employee(employees):
    name = input("Enter the name of the employee to update: ").strip()
    for emp in employees:
        if emp.get_name().lower() == name.lower():
            emp.set_age(int(input("Enter new age: ")))
            emp.set_salary(float(input("Enter new salary: ")))
            if isinstance(emp, Manager):
                emp.set_department(input("Enter new department: ").strip())
            elif isinstance(emp, Worker):
                emp.set_hours_worked(int(input("Enter new hours worked: ")))
            print("Employee information updated successfully.")
            return
    print("Employee not found.")



def delete_employee(employees):
    name = input("Enter the name of the employee to delete: ").strip()
    for emp in employees:
```

```
        if emp.get_name().lower() == name.lower():
            employees.remove(emp)
            print("Employee deleted successfully.")
            return
    print("Employee not found.")


filename = "employees.csv"
employees = load_from_csv(filename)
while True:
    print("\nEmployee Management System")
    print("1.Add the Employee")
    print("2.Display the Employees")
    print("3.Update the Employee")
    print("4.Delete the Employee")
    print("5.Exit")
    choice = input("Enter your choice : ").strip()

    if choice == "1":
        add_employee(employees)
    elif choice == "2":
        display_employees(employees)
    elif choice == "3":
        update_employee(employees)
    elif choice == "4":
        delete_employee(employees)
    elif choice == "5":
        save_to_csv(filename, employees)
        print("Exit the program")
        break
    else:
        print("Invalid choice Please enter teh valide number")
```