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 set\_name(self, name):

self.\_\_name = name

def get\_age(self):

return self.\_\_age

def set\_age(self, age):

self.\_\_age = age

def get\_salary(self):

return self.\_\_salary

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

class Employee:

def \_\_init\_\_(self, name, age, salary, department, hours\_worked):

self.name = name

self.age = age

self.salary = salary

self.department = department

self.hours\_worked = hours\_worked

import csv

def write\_to\_csv(file\_name, employees):

with open(file\_name, mode='w', newline='') as file:

writer = csv.writer(file)

writer.writerow(["Name", "Age", "Salary", "Department", "Hours Worked"])

for emp in employees:

writer.writerow([emp.name, emp.age, emp.salary, emp.department, emp.hours\_worked])

def read\_from\_csv(file\_name):

employees = []

with open(file\_name, mode='r') as file:

reader = csv.reader(file)

next(reader) # Skip header

for row in reader:

name, age, salary, department, hours\_worked = row

employees.append(Employee(name, int(age), float(salary), department, int(hours\_worked)))

return employees

def add\_employee(employee, employees):

employees.append(employee)

def display\_employees(employees):

for emp in employees:

print(f"Name: {emp.name}, Age: {emp.age}, Salary: {emp.salary}, Department: {emp.department}, Hours Worked: {emp.hours\_worked}")

def update\_employee(employee, employees):

for emp in employees:

if emp.name == employee.name:

emp.age = employee.age

emp.salary = employee.salary

emp.department = employee.department

emp.hours\_worked = employee.hours\_worked

def delete\_employee(name, employees):

employees[:] = [emp for emp in employees if emp.name != name]

employees = []

while True:

print("\nEmployee Information Management System")

print("1. Add Employee")

print("2. Display Employees")

print("3. Update Employee Information")

print("4. Delete Employee")

print("5. Exit")

choice = input("Enter your choice: ")

if choice == '1':

name = input("Enter Name: ")

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

salary = float(input("Enter Salary: "))

department = input("Enter Department: ")

hours\_worked = int(input("Enter Hours Worked: "))

new\_employee = Employee(name, age, salary, department, hours\_worked)

add\_employee(new\_employee, employees)

write\_to\_csv("employees.csv", employees)

elif choice == '2':

employees = read\_from\_csv("employees.csv")

display\_employees(employees)

elif choice == '3':

name = input("Enter Name of Employee to Update: ")

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

salary = float(input("Enter New Salary: "))

department = input("Enter New Department: ")

hours\_worked = int(input("Enter New Hours Worked: "))

updated\_employee = Employee(name, age, salary, department, hours\_worked)

update\_employee(updated\_employee, employees)

write\_to\_csv("employees.csv", employees)

elif choice == '4':

name = input("Enter Name of Employee to Delete: ")

delete\_employee(name, employees)

write\_to\_csv("employees.csv", employees)

elif choice == '5':

break

else:

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