Ex.No.: 8 20.03.25	INHERITANCE	Register Number: URK24CS1189

Aim:-

To create a base class called Employee and derive sub classes , create a class Worker and derive two classes DailyWorker and SalariedWorker.

1. Develop a python application using Inheritance concept to automate the salary calculation of employee in an organization as per the salary band given below. Create a base class called Employee and derive sub classes as per the given table. Apply method overriding to implement the following services via menu driven interface.
a) Calculate Gross Salary
b) Calculate Net Salary
c) Calculate Tax
d) Print the Pay Details

SalaryBand	Manager	Engineer
BasicSalary	30000	20000
DAPay	95%	80%
HRA	20%	15%
TAX	25%	15%
EPF	3000	2000

Algorithm:-

Step 1: Start.
Step 2: Create a class named ‘Employee’ with four functions named ‘calculate_gross_salary’ ‘calculate_tax’ , ‘calculate_net_salary’ , and ‘print_pay_details’ for calculating the salary of employee.
Step 3: Create a derive class name ‘Manager’ and ‘Engineer’ , get the data from ‘employee’.
Step 4: Using while loop , check the choice for choose the Employee type and the function to calculate.
Step 5: In function ‘calculate_gross_salary’ , da = (da / 100) * basic_salary and
hra = (hra / 100) * basic_salary.
Step 6: In function ‘calculate_tax’ , (tax /100)* gross_salary and In function ‘calculate_net_salary’ , gross_salary - tax_epf.
Step 7: In function ‘print_pay_details’ , Used to print all the data.
Step 8: End.

Program:-

```
Exp 8.py > ...
    URK24CS1189
    class Employee:
        def __init__(self,basic_salary,da_percent,hra_percent,tax_percent,epf):
            self.basic_salary=basic_salary
            self.da_percent=da_percent
            self.hra_percent=hra_percent
            self.tax_percent=tax_percent
            self.epf=epf
        def calculate_gross_salary(self):
            da=(self.da_percent/100)*self.basic_salary
            hra=(self.hra_percent/100)*self.basic_salary
            return self.basic_salary+da+hra
        def calculate_tax(self):
            return(self.tax_percent/100)*self.calculate_gross_salary()
        def calculate_net_salary(Self):
            return Self.calculate_gross_salary()-Self.calculate_tax()-Self.epf
        def print_pay_details(self):
            print(f"Basic salary:{self.basic_salary}")
            print(f"DA:{self.da_percent}%")
            print(f"HRA:{self.hra_percent}%")
            print(f"Gross Salary:{self.calculate_gross_salary()}")
            print(f"Tax:{self.calculate_tax()}")
            print(f"EPF:{self.epf}")
            print(f"Net Salary:{self.calculate_net_salary()}\n")
    class Manager(Employee):
        def __init__(self):
            super().__init__(basic_salary=30000,da_percent=95,hra_percent=20,tax_percent=25,epf=3000)
    class Engineer(Employee):
        def __init__(self):
            super().__init__(basic_salary=20000,da_percent=80,hra_percent=15,tax_percent=15,epf=2000)
    while True:
        print("Choose an employee type:")
        print("1. Manager","\n2.Engineer","\n3.Exit")
        choice=int(input("Enter your choice:"))
        if choice=="1":
            employee=Manager()
        elif choice=="2":
            employee==Engineer()
        elif choice=="3":
            break
        else:
            print("Invalid choice! Try Again.")
            continue
        print("Choose an optio:")
        print("a) calculate Gross salary")
        print("b)calculate Net Salary")
        print("c)calculate Tax")
        print("d)print pay details")
        option=input("Enter an option: ").lower()
        if option=="a":
            print(f"Gross Salary:{employee.calculate_gross_salary()}\n")
        elif option=="b":
            print(f"Net Salary:{employee.calculate_net_salary()}\n")
        elif option=="c":
            print(f"Tax:{employee.calculate_tax()}\n")
        elif option=="d":
            employee.print_pay_details()
        else:
            print("Invalid option! Try again.")
```

Output:-

```
Choose an Employee Type:
1. Manager
2. Engineer
3. Exit
Enter choice: 1
Choose an option:
a) Calculate Gross Salary
b) Calculate Net Salary
c) Calculate Tax
d) Print Pay Details
Enter option: d
Basic Salary: 30000
DA: 95%
HRA: 20%
Gross Salary: 64500.0
Tax: 16125.0
EPF: 3000
Net Salary: 45375.0

Choose an Employee Type:
1. Manager
2. Engineer
3. Exit
Enter choice: 2
Choose an option:
a) Calculate Gross Salary
b) Calculate Net Salary
c) Calculate Tax
d) Print Pay Details
Enter option: d
Basic Salary: 20000
DA: 80%
HRA: 15%
Gross Salary: 39000.0
Tax: 5850.0
EPF: 2000
Net Salary: 31150.0

Choose an Employee Type:
1. Manager
2. Engineer
3. Exit
Enter choice: 3
```

2. Develop a python application using Inheritance as per the following. Create a class Worker and derive two classes DailyWorker and SalariedWorker from it. Every worker has name, salary rate. Provide a method Com_Pay (int hours) to compute the week pay of every worker. A DailyWorker is paid on the basis of number of days he/she works. The SalariedWorker gets paid the wage for 40 hours a week no matter what actual hours is. Implement this scenario to calculate the pay of workers.

Algorithm:-

Step 1: Start.
Step 2: Create a base class ‘Worker’ with two function ‘comp_pay’ and ‘display’.
Step 3: Create a derive class ‘DailyWorker’ and ‘SalariedWork’ with function ‘comp_pay’ , get the data from ‘Worker’.
Step 4: In ‘comp_pay’ function , salary_rate * hours and In ‘display’ function , it display the name and salary_rate.
Step 5: Create a objects for two derive class for display the details about workers using ‘display’ function and ‘comp_pay’ function.
Step 6: End.

Program:-

```
#URK24CS1189
class Worker:
    def _init_(self, name, salary_rate):
        self.name = name
        self.salary_rate = salary_rate
    def comp_pay(self, hours):
        pass
    def display(self):
        print(f"Worker Name: {self.name}")
        print(f"Salary Rate: {self.salary_rate}")
class DailyWorker(Worker):
    def comp_pay(self, hours):
        return self.salary_rate * hours
class SalariedWorker(Worker):
    def comp_pay(self, hours):
        return self.salary_rate * 40
worker1 =DailyWorker("John", 500)
worker2 = SalariedWorker("Alice", 400)
print("Daily Worker Details:")
worker1.display()
print(f"Weekly Pay (worked 6 days, 8 hours/day): {worker1.comp_pay(6 * 8)}\n")
print("Salaried Worker Details:")
worker2.display()
print(f"Weekly Pay (worked 45 hours but fixed pay for 40 hours): {worker2.comp_pay(45)}\n")
```

Output:-

```
Daily Worker Details:-
Worker Name: John
Salary Rate: 500
Weekly Pay (worked 6 days, 8 hours/day): 24000

Salaried Worker Details:-
Worker Name: Alice
Salary Rate: 400
Weekly Pay (worked 45 hours but fixed pay for 40 hours): 16000
```

Result:-

Thus the all program using inheritance has been run successfully.