**ASSINGMENT 3**

**Name: Manasa Vathumilli**

**Id: 700745467**

1. **Create a class Employee and then do the following.**

**• Create a data member to count the number of Employees**

**• Create a constructor to initialize name, family, salary, department**

**• Create a function to average salary**

**• Create a Fulltime Employee class and it should inherit the properties of Employee class**

**• Create the instances of Fulltime Employee class and Employee class and call their member functions.**

class Employee:

# Creating a data member to count the number of Employees

no\_of\_employees = 0

# Initializing name, family, salary, department

def \_\_init\_\_(self, name, family\_name, salary, department):

self.\_\_name = name

self.\_\_family\_name = family\_name

self.salary = salary

self.\_\_department = department

Employee.no\_of\_employees += 1

#Creating a function to average salary

@staticmethod

def average\_salary(employees):

sum = 0

for employee in employees:

sum += employee.salary

return sum / Employee.no\_of\_employees

#Creating a Fulltime Employee class

class FulltimeEmployee(Employee):

def \_\_init\_\_(self, name, family\_name, salary, department):

super().\_\_init\_\_(name, family\_name, salary, department)

def full\_time\_benefits(self):

print("Few benefits as full time employee.")

# Creating the instances of Fulltime Employee class and Employee class and also calling their member functions.

def main():

employees = []

fte1 = FulltimeEmployee("Employee1", "FamilyName1", 120000, "Management")

fte1.full\_time\_benefits()

employees.append(fte1)

fte2 = FulltimeEmployee("Employee2", "FamilyName2", 180000, "RnD")

employees.append(fte2)

emp1 = Employee("Employee3", "FamilyName3", 160000, "Marketing")

employees.append(emp1)

emp2 = Employee("Employee4", "FamilyName4", 135000, "HR")

employees.append(emp2)

print("Average salary:", FulltimeEmployee.average\_salary(employees))

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Description:**

Here I created a class for employee after that initialized name ,family,salary,department using And also called static method, to find out the average salary I created function to get it the average salary. And created full time employee class to nherit the properties of employe class created above. At last I created a main class to call all its member functions.

**Screenshot of source code and output:**

Text

Description automatically generatedA picture containing application

Description automatically generated

1. **NumPy**

**Using NumPy create random vector of size 20 having only float in the range 1-20. Then reshape the array to 4 by 5 Then replace the max in each row by 0 (axis=1) (you can NOT implement it via for loop)**

import numpy as np

#Creating a random vector of size 20

v = np.random.uniform(1, 20, 20)

v = v. reshape(4, 5)

# Reshape the array to 4 by 5

print(v)

v[np.arange(4), v.argmax(axis=1)]=0

# Replace the max value with 0 (axis=1)

print(v)

**Description:** In the below program I created a random vector of size 20 and using reshape function to reshape the array and used arrange and arg max functions to replace it with value 0.

**Screenshot of source code and output:**

Graphical user interface, text, application

Description automatically generated

**Video Link:**

**GitHub:** https://github.com/Manasav17/NNDL-ICP-3

Source Code: http://localhost:8889/notebooks/Documents/Neural/NNDL%20ICP%203.ipynb