**ICP-2 Bhavana Parasa(700761185)**

[**https://github.com/BXP11850/Neural-Networks-Deep-Learning-Assignments-BXP11850.git**](https://github.com/BXP11850/Neural-Networks-Deep-Learning-Assignments-BXP11850.git)

**QUESTION :-** 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

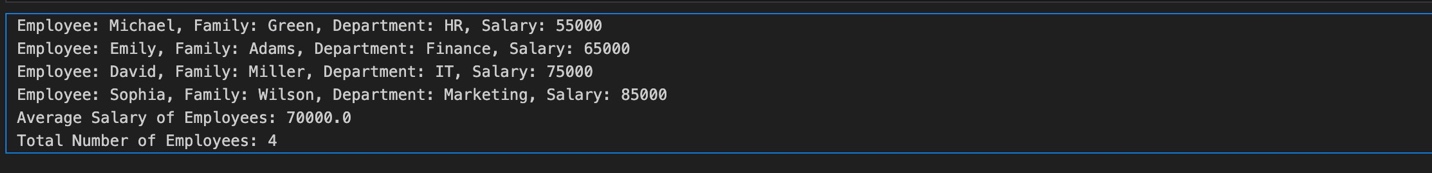
• Create the instances of Fulltime Employee class and Employee class and call their member functions

**CODE :-**

A computer screen shot of a program code

Description automatically generated

**OUTPUT :-**



This code defines two classes, `Employee` and `FulltimeEmployee`.

**- Employee Class:**

- Tracks the number of employees and their total salary.

- Initializes each employee's name, family, salary, and department.

- Provides methods to calculate the average salary and display employee details.

**- FulltimeEmployee Class:**

- Inherits from `Employee` and uses the same properties and methods.

**Example Use:**

- Creates four employees (two general and two full-time).

- Displays each employee's information.

- Calculates and prints the average salary across all employees.

- Displays the total number of employees.

**CODE- 2**

2. 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)

A screenshot of a computer program

Description automatically generated

**OUTPUT :-**

A screenshot of a computer

Description automatically generated

**Explanation:**

1. **Creating a random vector**: We use np.random.uniform(1, 20, 20) to generate 20 random floats between 1 and 20.

2. **Reshaping the array**: The reshape(4, 5) method reshapes the 20-element array into a 4x5 array.

3. **Replacing the max value in each row**: We calculate the maximum value in each row with np.max(reshaped\_array, axis=1, keepdims=True). Then, we use boolean indexing to replace these max values with 0.