

# Spring 2024: CS5720 Neural Networks & Deep Learning -

## ICP-3Assignment-3

Name: Pushkara Naga Sai Sri Vyshnavi Chakka

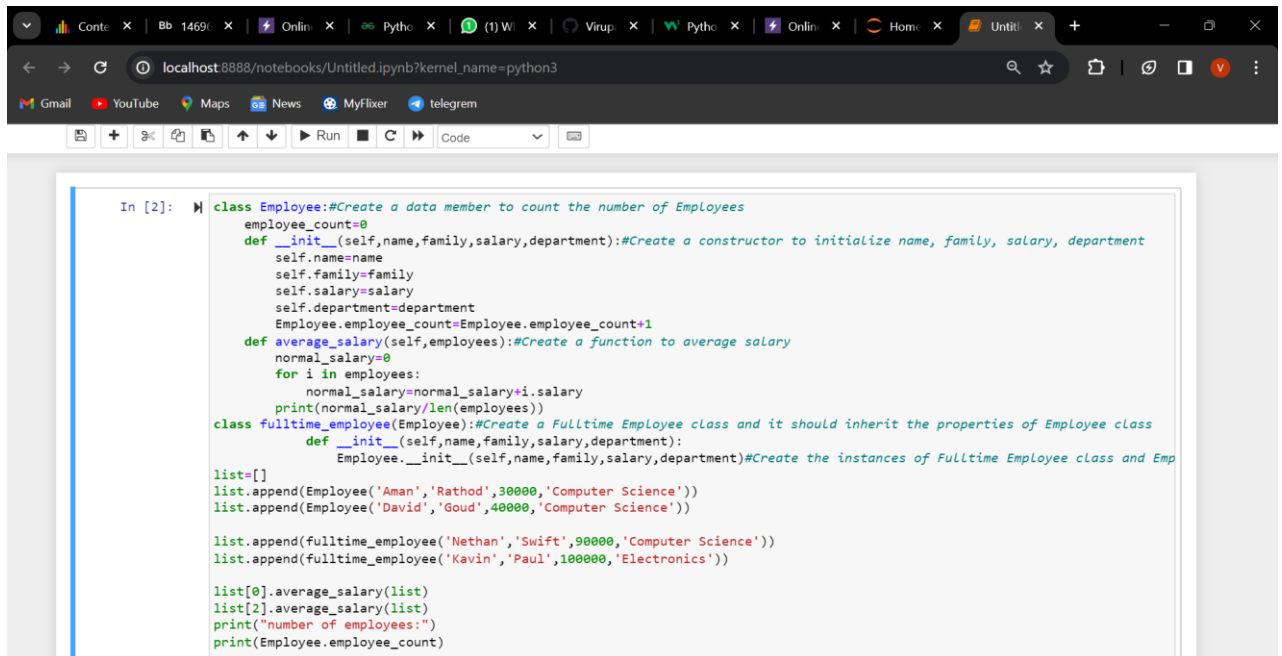
STUDENT ID:700752861

Github link: [https://github.com/PushkaraChakka/Assignment\\_3\\_icp3](https://github.com/PushkaraChakka/Assignment_3_icp3)

Video link:

[https://drive.google.com/file/d/1sDyo5xLGp3S2Dm1caC3w7\\_zW3kiw5Yok/view?usp=sharing](https://drive.google.com/file/d/1sDyo5xLGp3S2Dm1caC3w7_zW3kiw5Yok/view?usp=sharing)

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.



```
In [2]: class Employee:#Create a data member to count the number of Employees
        employee_count=0
        def __init__(self,name,family,salary,department):#Create a constructor to initialize name, family, salary, department
            self.name=name
            self.family=family
            self.salary=salary
            self.department=department
            Employee.employee_count=Employee.employee_count+1
        def average_salary(self,employees):#Create a function to average salary
            normal_salary=0
            for i in employees:
                normal_salary=normal_salary+i.salary
            print(normal_salary/len(employees))
        class fulltime_employee(Employee):#Create a Fulltime Employee class and it should inherit the properties of Employee class
            def __init__(self,name,family,salary,department):
                Employee.__init__(self,name,family,salary,department)#Create the instances of Fulltime Employee class and Emp

list=[]
list.append(Employee('Aman','Rathod',30000,'Computer Science'))
list.append(Employee('David','Goud',40000,'Computer Science'))

list.append(fulltime_employee('Nethan','Swift',90000,'Computer Science'))
list.append(fulltime_employee('Kavin','Paul',100000,'Electronics'))

list[0].average_salary(list)
list[2].average_salary(list)
print("number of employees:")
print(Employee.employee_count)
```

Output:

```
65000.0
65000.0
number of employees:
4
```

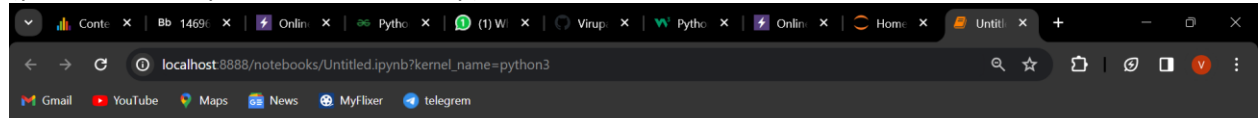
## 2. Numpy

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)



65000.0

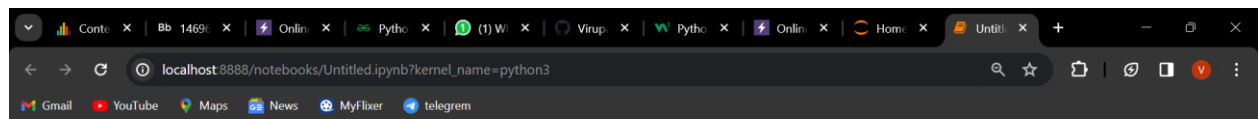
number of employees:

4

```
[3]: ▶ import numpy as np
      # Create random vector of size 20 with floats between 1 and 20
      vec = np.random.uniform(9, 6, 20)
      # Reshape the vector to 4 by 5
      mat = vec.reshape(4, 5)
      # Replacing the max in each row by 0
      mat[np.arange(4), mat.argmax(axis=1)] = 0
      # Print the output
      print(mat)
```

```
[[6.16609835 7.57934912 8.59580213 0.          8.47324501]
```

Output:



```
# Create random vector of size 20 with floats between 1 and 20
vec = np.random.uniform(9, 6, 20)
# Reshape the vector to 4 by 5
mat = vec.reshape(4, 5)
# Replacing the max in each row by 0
mat[np.arange(4), mat.argmax(axis=1)] = 0
# Print the output
print(mat)
```

```
[[6.16609835 7.57934912 8.59580213 0.          8.47324501]
 [6.37335039 8.62410654 6.59939819 7.50061315 0.         ]
 [6.40314712 7.12251805 0.          7.26664749 7.72637134]
 [8.05537403 6.46703327 8.15758698 6.73592106 0.         ]]
```

In [ ]: ▶

---

i