

# NNDL\_ICP3

STUDENT NAME: Mounika Chigurupati

STUDENT ID: 700749935

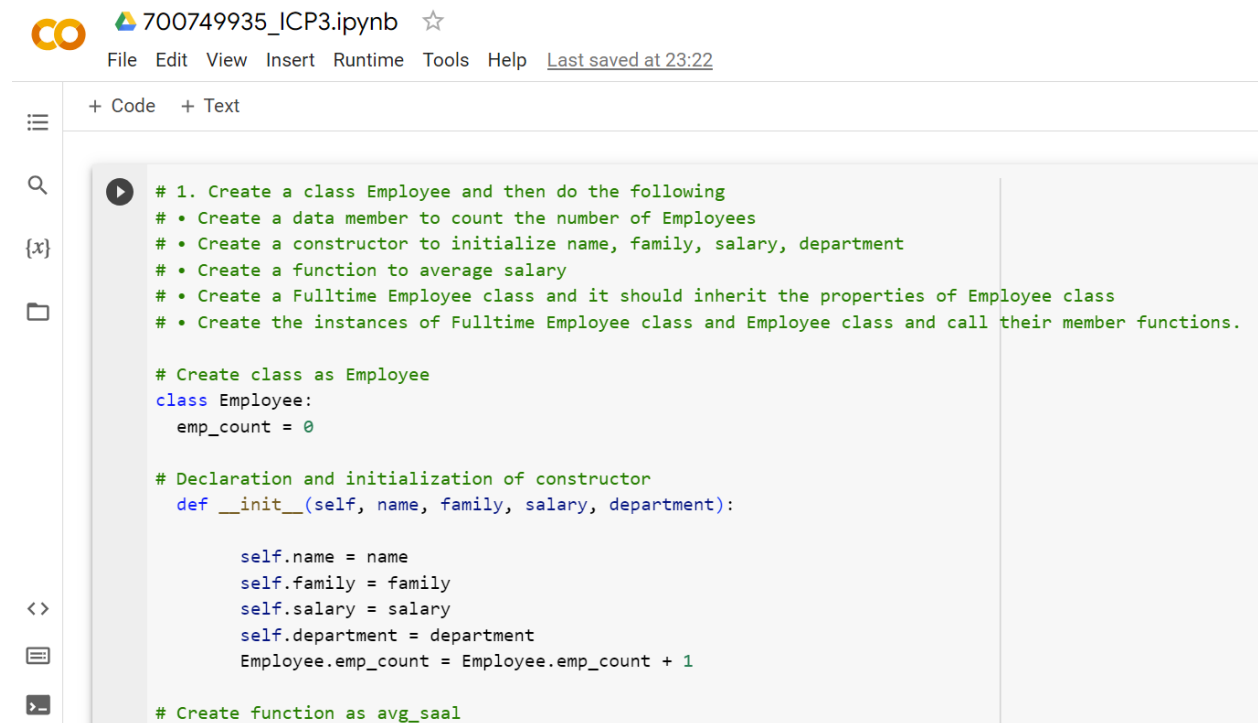
GitHub Link: <https://github.com/Mounika112121/Assignment-3-ICP>

Video Link:

<https://drive.google.com/file/d/189jBVJwoy02BFsopVa30GnIVaAgcpktA/view?usp=sharing>

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 members functions.



The screenshot shows a Jupyter Notebook titled "700749935\_ICP3.ipynb". The code is as follows:

```
# 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.

# Create class as Employee
class Employee:
    emp_count = 0

# Declaration and initialization of constructor
    def __init__(self, name, family, salary, department):

        self.name = name
        self.family = family
        self.salary = salary
        self.department = department
        Employee.emp_count = Employee.emp_count + 1

# Create function as avg_saal
```



700749935\_ICP3.ipynb ☆

File Edit View Insert Runtime Tools Help Last saved at 17:13

+ Code + Text



{x}



&lt;&gt;



```
def avg_sal(self, emps):
    sum_sal = 0
    for i in emps:
        sum_sal= sum_sal+ i.salary

    # Print output
    print(sum_sal/len(emps))

    # Create class as Fulltime_Employee
    class Fulltime_Employee(Employee):

        def __init__(self, name, family, salary, department):
            Employee.__init__(self, name, family, salary, department)

    list = []
    list.append(Employee('Jamielannister', 'Joffrey', 30000, 'Accounts'))
    list.append(Employee('Branstark', 'Emiliaclarke', 45000, 'Salesforce'))

    list.append(Fulltime_Employee('Johnsnow', 'Arrya', 23000, 'Architectutre'))
    list.append(Fulltime_Employee('Sophie', 'Robb', 50000, 'Management'))

    list[0].avg_sal(list)
    list[2].avg_sal(list)
```



700749935\_ICP3.ipynb ☆

File Edit View Insert Runtime Tools Help Last saved at 17:13

+ Code + Text



{x}



```
# Print output as employee count
print(Employee.emp_count)
```

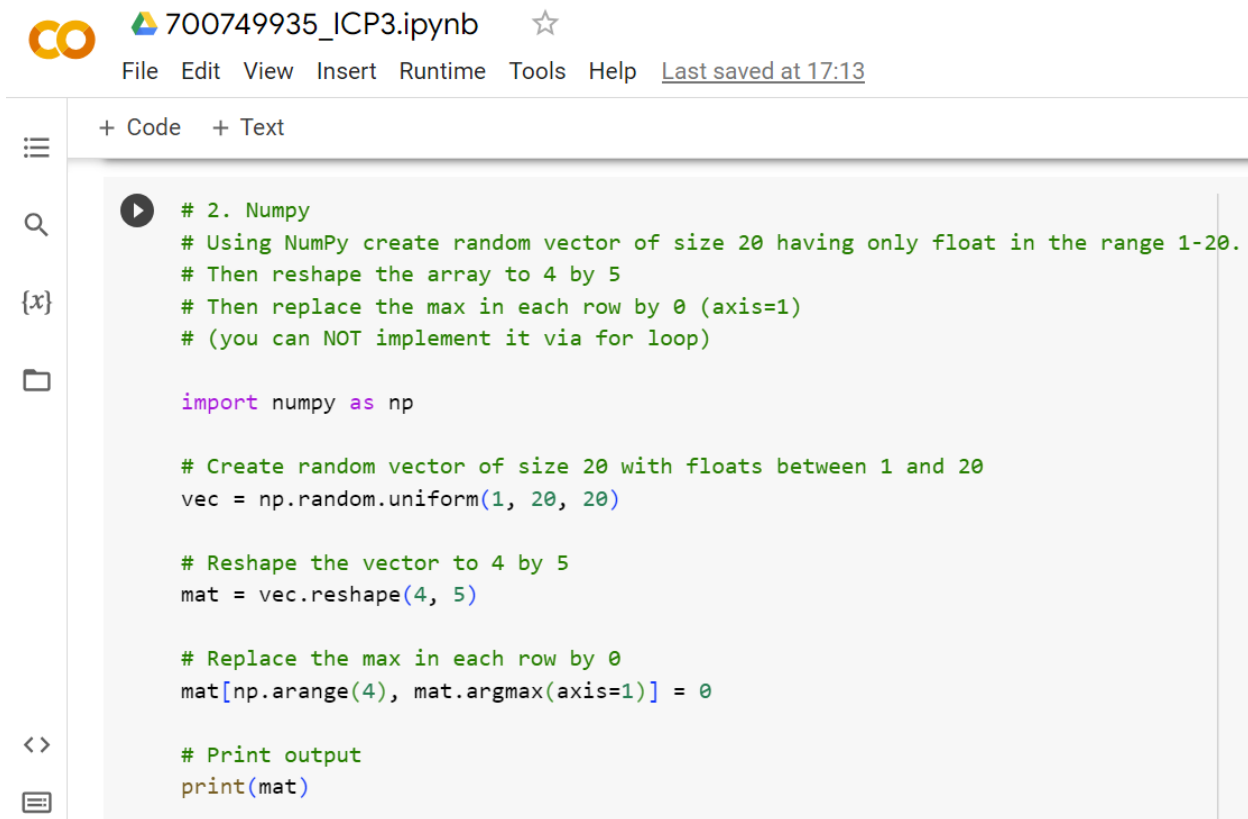
```
37000.0
37000.0
4
```

## 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)



CO 700749935\_ICP3.ipynb ☆

File Edit View Insert Runtime Tools Help Last saved at 17:13

+ Code + Text

```
# 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)

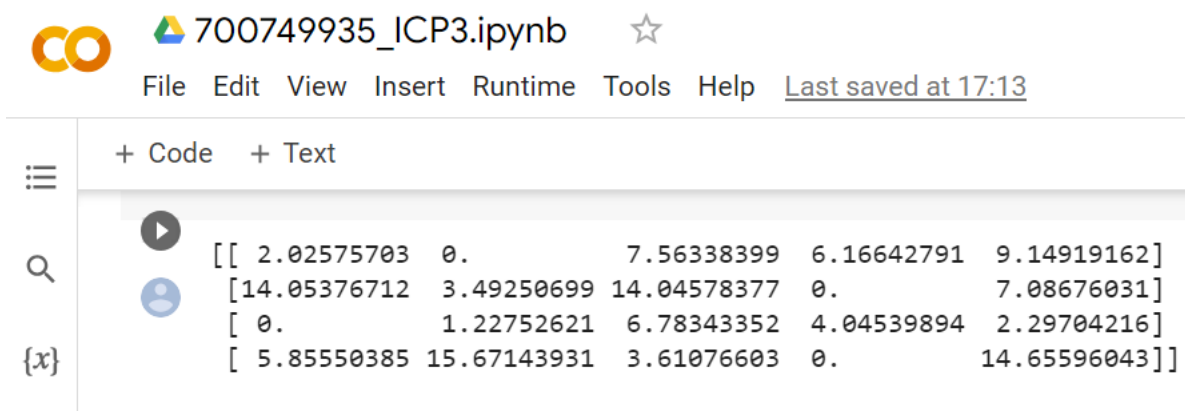
import numpy as np

# Create random vector of size 20 with floats between 1 and 20
vec = np.random.uniform(1, 20, 20)

# Reshape the vector to 4 by 5
mat = vec.reshape(4, 5)

# Replace the max in each row by 0
mat[np.arange(4), mat.argmax(axis=1)] = 0

# Print output
print(mat)
```



CO 700749935\_ICP3.ipynb ☆

File Edit View Insert Runtime Tools Help Last saved at 17:13

+ Code + Text

```
[[ 2.02575703  0.          7.56338399  6.16642791  9.14919162]
 [14.05376712  3.49250699 14.04578377  0.          7.08676031]
 [ 0.          1.22752621  6.78343352  4.04539894  2.29704216]
 [ 5.85550385 15.67143931  3.61076603  0.          14.65596043]]
```