**Q1 :**

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

* Create a data member to count the number of Employees

class Employee:

no\_of\_employees = 0

* Create a constructor to initialize 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

* Create a function to average salary

@staticmethod

def average\_salary(employees):

"""

function to average salary

"""

sum = 0

for employee in employees:

sum += employee.salary

return sum / Employee.no\_of\_employees

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

class FulltimeEmployee(Employee):

"""

Full Time Employee is a sub class of 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.")

* Create the instances of Fulltime Employee class and Employee class and call 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()

Text

Description automatically generated

Q2 :

**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)

import numpy as np

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

print(a)

r = np.reshape(a, (4,5))

max\_index = np.argmax(r, axis = 1)

r[np.arange(len(max\_index)), max\_index] = 0

print(r)

Graphical user interface, text

Description automatically generated

Video Link:

<https://drive.google.com/file/d/1yjxO_DCmBmt9kuX8A1vljQhnFu9b1wG9/view?usp=sharing>

Git hub: