# 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:

employee\_count = 0

# initialization and declaration of constructor

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

self.name = name

self.family = family

self.salary = salary

self.department = department

Employee.employee\_count = Employee.employee\_count + 1

# Creating function as avg\_saal

def avg\_sal(self, employees):

sum\_sal = 0

for i in employees:

sum\_sal= sum\_sal+ i.salary

# Printing output

print(sum\_sal/len(employees))

# 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('Nick', 'Jones', 20000, 'hero'))

list.append(Employee('Brad', 'henry', 35000, 'DOP'))

list.append(Fulltime\_Employee('alex', 'hales', 25000, 'director'))

list.append(Fulltime\_Employee('stefhen', 'marek', 40000, 'producer'))

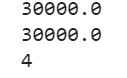
list[0].avg\_sal(list)

list[2].avg\_sal(list)

# Print output as employee count

print(Employee.employee\_count)

Output:



# 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

# Creating 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)

# Replacing the max in each row by 0

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

# Print the output

print(mat)

Output:

