**UshaKiran Yadav Avula**

**700746114**

**ICP3**

**Program1:**

# Created Employee class with name, family, salary and department

class Employee:

    # declared a data member to count the number of Employees

    no\_of\_employees = 0

    # constructor to initialize the object's attributes

    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

    # declared average\_salary function to return average salary

    def average\_salary(employees):

        """

        function to average salary

        """

        sum = 0

        for employee in employees:

            sum += employee.salary

        return sum / Employee.no\_of\_employees

# Created a Fulltime Employee class and inherited 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\_member(self):

        print("Calling FulltimeEmployee member function.")

# Created the instances of Fulltime Employee class and Employee class and calling their member functions.

def main():

    employees = []

    ftEmployee1 = FulltimeEmployee("Vamshi", "Ponugoti", 140000, "Software Engineering")

    ftEmployee1.full\_time\_member()

    employees.append(ftEmployee1)

    ftEmployee2 = FulltimeEmployee("Micheal", "Velayudham", 170000, "Cyber Security")

    employees.append(ftEmployee2)

    employee1 = Employee("CoseC", "Pacchipulsula", 150000, "Testing")

    employees.append(employee1)

    employee2 = Employee("VeeraSimhaReddy", "Nandamuri", 192000, "Product Manager")

    employees.append(employee2)

    print("Average salary:", FulltimeEmployee.average\_salary(employees))

    print("total no of employees:",Employee.no\_of\_employees)

# declared this method to execute code When the file runs as a script.

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

**A screenshot of a computer

Description automatically generated**

**Program2:**

import numpy as np

import numpy as np

# created a random vector of size 20 with float values between 1 and 20

ranvec = np.random.uniform(low=1, high=20, size=20)

# reshape the array to 4 by 5 using reshape method

mat45 = ranvec.reshape(4, 5)

print(mat45)

print("\n Replacing the maximum in each row by 0 using where method\n")

# replace the max in each row by 0 using where method

mat45 = np.where(mat45 == np.amax(mat45, axis=1, keepdims=True), 0, mat45)

print(mat45)

**Output:**

**A screenshot of a computer

Description automatically generated**

**GitHub:**