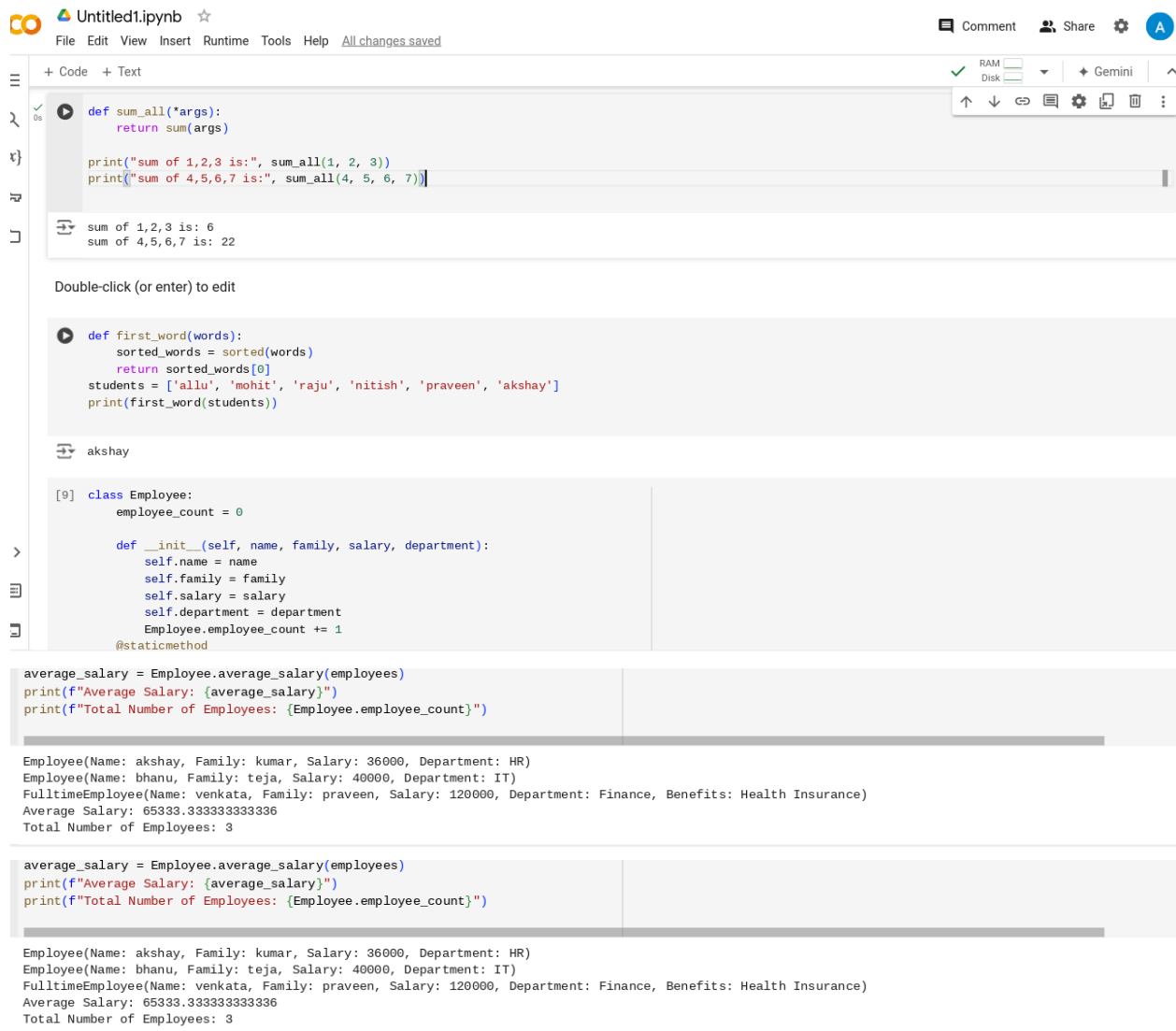


ICP2 REPORT

1.Explain the difference between Counter.count and self._count Counter.count :Scope: Shared across all instances of the Counter class.Purpose: Keeps track of a value common to all instances. In this case, it tracks the total number of increments made by all instances.Access: Can be accessed using Counter.count or any instance (instance.count), but modifying it affects all instances.self._count (Instance Variable):Scope: Specific to the individual instance.Purpose: Keeps track of a value unique to that instance. In this case, it tracks the number of increments made to that particular instance.Access: Accessed using self within instance methods.

2.What is the output of a.get_counts() and b.get_counts() a.get_counts():Returns: "Instance count: 2, Class count: 3" b.get_counts():Returns: "Instance count: 1, Class count: 3"

3"How does the increment method affect both the class and instance variables?Instance Variable:Effect: Increases by 1 each time the increment method is called on that instance. This reflects the count of increments specific to that instance.Class Variable (Counter.count):**Effect:** Also increases by 1 each time the increment method is called on any instance. This reflects the total number of increments across all instances.while **Counter.count** tracks the total number of increments across all instances.The increment method modifies both variables, demonstrating how instance and class variables interact in Python classes



```
def sum_all(*args):
    return sum(args)

print("sum of 1,2,3 is:", sum_all(1, 2, 3))
print("sum of 4,5,6,7 is:", sum_all(4, 5, 6, 7))
```

```
sum of 1,2,3 is: 6
sum of 4,5,6,7 is: 22
```

```
Double-click (or enter) to edit
```

```
def first_word(words):
    sorted_words = sorted(words)
    return sorted_words[0]

students = ['allu', 'mohit', 'raju', 'nitish', 'praveen', 'akshay']
print(first_word(students))
```

```
akshay
```

```
[9] class Employee:
    employee_count = 0

    def __init__(self, name, family, salary, department):
        self.name = name
        self.family = family
        self.salary = salary
        self.department = department
        Employee.employee_count += 1

    @staticmethod
```

```
average_salary = Employee.average_salary(employees)
print(f"Average Salary: {average_salary}")
print(f"Total Number of Employees: {Employee.employee_count}")
```

```
Employee(Name: akshay, Family: kumar, Salary: 36000, Department: HR)
Employee(Name: bhanu, Family: teja, Salary: 40000, Department: IT)
FulltimeEmployee(Name: venkata, Family: praveen, Salary: 120000, Department: Finance, Benefits: Health Insurance)
Average Salary: 65333.333333333336
Total Number of Employees: 3
```

```
average_salary = Employee.average_salary(employees)
print(f"Average Salary: {average_salary}")
print(f"Total Number of Employees: {Employee.employee_count}")
```

```
Employee(Name: akshay, Family: kumar, Salary: 36000, Department: HR)
Employee(Name: bhanu, Family: teja, Salary: 40000, Department: IT)
FulltimeEmployee(Name: venkata, Family: praveen, Salary: 120000, Department: Finance, Benefits: Health Insurance)
Average Salary: 65333.333333333336
Total Number of Employees: 3
```

MY GITHUB LINK:<https://github.com/akshaykumarpatham/bda.git>