```
[ ]  Start coding or generate with AI.
```

The Difference between the Counter.count and self._count:

Counter.count

It is a class variable which tracks all the increments from all the objects. From the given code the total increments is 3.

Self._count:

It is a instance variable and it tracks Increments for one object. from the given code :2 for a, 1 for b.

The output of a.get_counts() and b.get_counts()?

a.get_counts(): Instance count: 2, Class count: 3

b.get_counts(): Instance count: 1, Class count: 3

How does the increment method affect the variables

When it comes to the class variables it effects all the objects i.e.,Increases for all objects. When it comes to the instance variables it applies to only one particular object.i.e,Increases only for that object.

icp2.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text Run all

RAM Disk

```
Start coding or generate with AI.
```

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

```
[ ] def first_word(word_list):
        return sorted(word_list)[0]
    students = ['Mary', 'Zelda', 'Jimmy', 'Jack', 'Bartholomew', 'Gertrude']
    print(first_word(students))
```

```
Bartholomew
```

```
0s
class Employee:
    count = 0
    total_salary = 0

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

        Employee.count += 1
        Employee.total_salary += salary

    @classmethod
    def average_salary(cls):
        return cls.total_salary / cls.count if cls.count else 0

class FulltimeEmployee(Employee):
```

Variables Terminal 11:12 PM Pyth

```
            self.department = department

            Employee.count += 1
            Employee.total_salary += salary

        @classmethod
        def average_salary(cls):
            return cls.total_salary / cls.count if cls.count else 0

    class FulltimeEmployee(Employee):
        pass

    # Create instances
    e1 = Employee("Amar", "Sai", 60000, "HR")
    e2 = FulltimeEmployee("Raj", "Johnson", 75000, "Engineering")
    e3 = FulltimeEmployee("Charlie", "Brown", 85000, "Marketing")

    # Outputs
    print("Total Employees:", Employee.count)
    print("Average Salary:", Employee.average_salary())
```

```
Total Employees: 3
Average Salary: 73333.33333333333
```