ICP 2 Report

1. This is a class variable. It is shared among all instances of the Counter class. Any change to Counter.count will be reflected across all instances of the class because there is only one copy of this variable, which is accessed by the class itself and by all instances. self._count: This is an instance variable. Each instance of the Counter class has its own separate copy of self._count. Changes to self._count only affect the specific instance in which the change is made.
2. What is the output of a.get_counts() and b.get_counts()? Let's break down the operations step by step:

When a = Counter() is called, an instance a is created with self._count = 0. When b = Counter() is called, another instance b is created with self._count = 0. Calling a.increment() twice increments a._count to 2 and Counter.count to 2. Calling b.increment() once increments b._count to 1 and Counter.count to 3. Now, a.get_counts() will return "Instance count: 2, Class count: 3", and b.get_counts() will return "Instance count: 1, Class count: 3".

3. The increment method increases the instance variable self._count by 1, affecting only the specific instance on which the method is called. The method also increases the class variable Counter.count by 1, affecting all instances of the class, as this variable is shared among them.

```
[16] # Mount Google Drive
     from google.colab import drive
     drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
def first_word(words):
    # Sort the list in alphabetical order
    sorted_words = sorted(words)

    # Return the first element in the sorted list
    return sorted_words[0]

# Example usage
students = ['Mary', 'Zelda', 'Jimmy', 'Jack', 'Bartholomew', 'Gertrude']
print(first_word(students))  # Output: 'Bartholomew'
```

```
Bartholomew
```

```
# This will overwrite the built-in sum function
def sum(*args):
    # some code
    pass

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: None
Sum of 4, 5, 6, 7 is: None
```

```python
class Employee:
    # Class-level attribute to count the number of Employees
    employee_count = 0
    total_salary = 0  # To keep track of the total salary for averaging

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

        # Update the class-level attribute for employee count and total salary
        Employee.employee_count += 1
        Employee.total_salary += salary

    @classmethod
    def average_salary(cls):
        if cls.employee_count == 0:
            return 0
        return cls.total_salary / cls.employee_count

    def display_info(self):
        return f"Name: {self.name}, Family: {self.family}, Salary: {self.salary}, Department: {self.department}"

class FulltimeEmployee(Employee):
    def __init__(self, name, family, salary, department, benefits):
        super().__init__(name, family, salary, department)
        self.benefits = benefits  # Additional attribute for FulltimeEmployee

    def display_info(self):
        # Include benefits in the display info
        basic_info = super().display_info()
        return f"{basic_info}, Benefits: {self.benefits}"

# Creating instances of Employee and FulltimeEmployee
```

```python
e1 = Employee("John Doe", "Smith", 50000, "Engineering")
e2 = Employee("Jane Roe", "Doe", 60000, "Marketing")
ft1 = FulltimeEmployee("Alice Johnson", "Brown", 70000, "HR", "Health Insurance")

# Calling member functions and displaying results
print(e1.display_info())
print(e2.display_info())
print(ft1.display_info())

# Display average salary
print(f"Average Salary: {Employee.average_salary()}")
```

```
Name: John Doe, Family: Smith, Salary: 50000, Department: Engineering
Name: Jane Roe, Family: Doe, Salary: 60000, Department: Marketing
Name: Alice Johnson, Family: Brown, Salary: 70000, Department: HR, Benefits: Health Insurance
Average Salary: 60000.0
```

Github Link
https://github.com/shashank1615/BDA.git