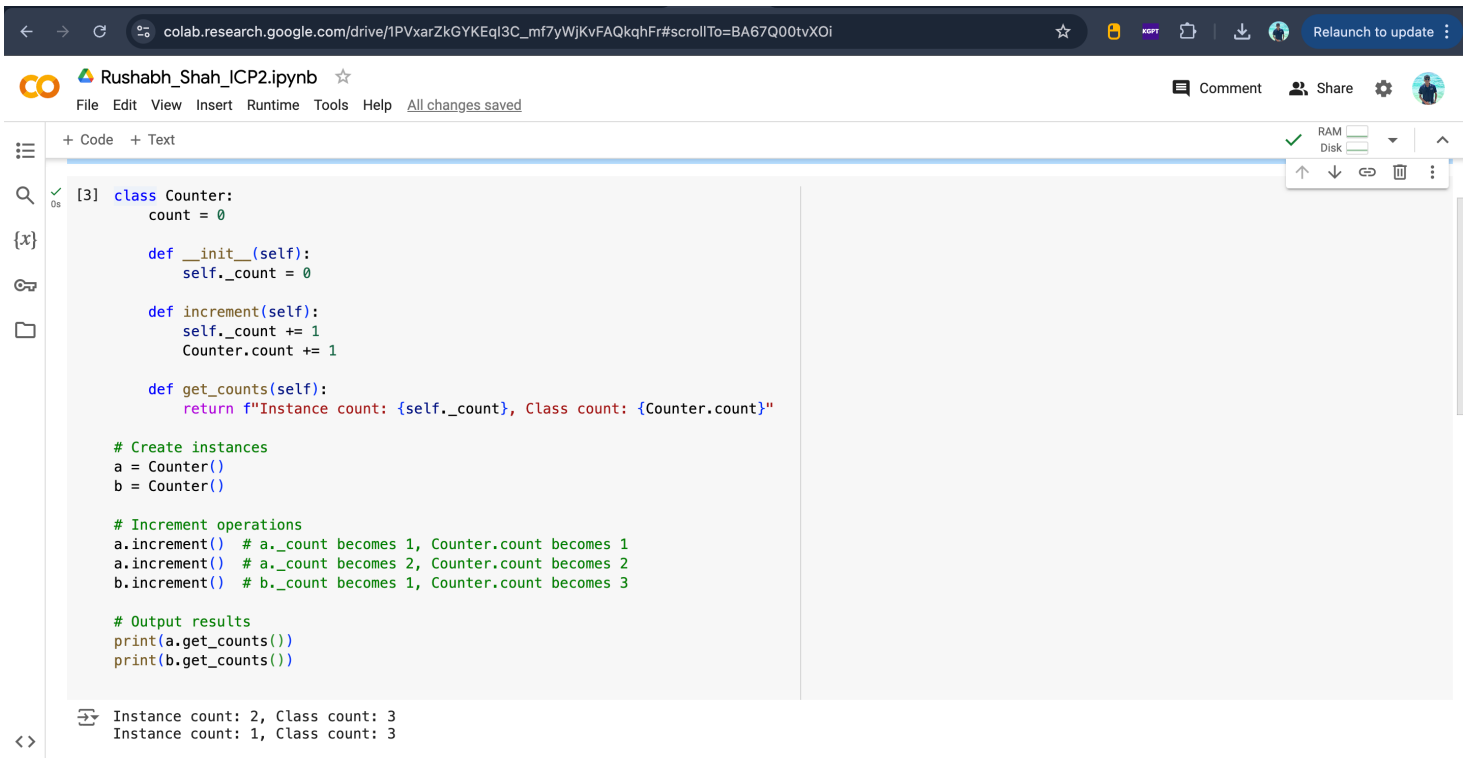


# Big data Analytics: ICP2

Answer: 1



The screenshot shows a Google Colab notebook titled "Rushabh\_Shah\_ICP2.ipynb". The code defines a `Counter` class with a class attribute `count = 0` and instance attributes `self._count = 0`. The `__init__` method initializes the instance count. The `increment` method increments both the instance and class counts. The `get_counts` method returns a string showing both counts. The code creates two instances, `a` and `b`, and performs increment operations. The output shows the counts for each instance and the total class count.

```
[3] class Counter:
    count = 0

    def __init__(self):
        self._count = 0

    def increment(self):
        self._count += 1
        Counter.count += 1

    def get_counts(self):
        return f"Instance count: {self._count}, Class count: {Counter.count}"

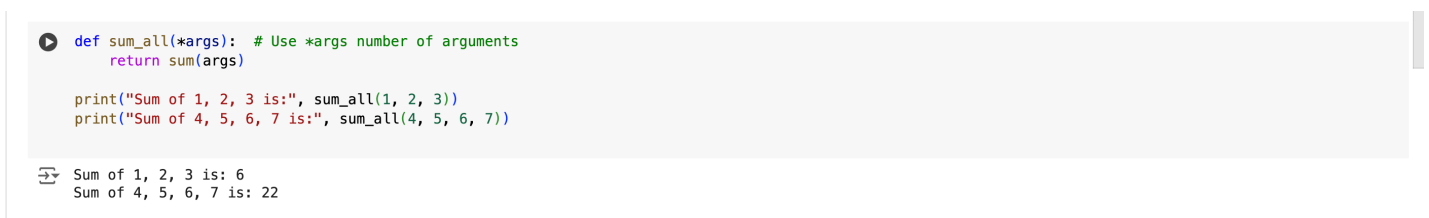
# Create instances
a = Counter()
b = Counter()

# Increment operations
a.increment() # a._count becomes 1, Counter.count becomes 1
a.increment() # a._count becomes 2, Counter.count becomes 2
b.increment() # b._count becomes 1, Counter.count becomes 3

# Output results
print(a.get_counts())
print(b.get_counts())
```

Instance count: 2, Class count: 3  
Instance count: 1, Class count: 3

Answer: 2



The screenshot shows a Google Colab notebook with a function `sum_all` that takes a variable number of arguments and returns their sum. The code prints the sum of 1, 2, and 3, and the sum of 4, 5, 6, and 7.

```
def sum_all(*args): # Use *args number of arguments
    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

Answer: 3



The screenshot shows a Google Colab notebook with a function `first_word` that takes a list of strings and returns the first word. The code creates a list of names and prints the first word.

```
def first_word(strings):
    return min(strings)

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

Bartholomew

Answer:4



```
+ Code + Text
class Employee:
    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.employee_count += 1
        Employee.total_salary += salary

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

class FulltimeEmployee(Employee):
    pass

# Create instances
emp1 = Employee("Mary", "Zelda", 30000, "HR")
emp2 = Employee("Jimmy", "Jack", 40000, "data analysis")
ft_emp1 = FulltimeEmployee("Bartholomew", "Gertrude", 50000, "Software developer")

# Print the results
print(f"Total Employees: {Employee.employee_count}")
print(f"Average Salary: ${Employee.average_salary():.2f}")

Total Employees: 3
Average Salary: $40000.00
```