

# ICP2 REPORT

A.

1. 'counter.count'(class variable):

this is a class level variable. it is shared across all instances of the 'counter' class. any changes to counter.count affect all instances of the class because it is a single variable shared by all objects of that class.

2. 'self.count'(instance variable):

this is an instance level variable. Each instance of the 'counter' class has its own separate '\_count'. changes to 'self.\_count' affect only the particular instance of the class.

B. 1. 'a.get\_counts()':

'a.count' is 2 (since 'a' was incremented twice) 'counter.count' is 3 (since both 'a' and 'b' have incremented the class variable)

so, the output is : instance count : 2 class count: 3

2. 'b.get\_counts()':

```
>'b._count' is 1 (since 'b' was incremented once).  
>'counter.count' is still 3( shared across all instances)
```

so the output is:

```
instance count : 1  
class count: 2
```

```
instance count : 1  
class count: 2
```

c. the increment method modifies both 'self.count' and 'counter.count'

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

> 'self.\_count+=1': this line increases the '\_count' for the instance that called the method. each instance has its own '\_count'

> 'counter.count+=1': this line increases the 'count' variable at the class level, which affects all instances of the 'counter' class.

```
[1] 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
```

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

```
Bartholomew
```

```
[3] 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
```

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

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

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

e1 = Employee("Alice", "Smith", 50000, "HR")
e2 = Employee("Bob", "Jones", 60000, "Engineering")
e3 = Employee("Charlie", "Brown", 55000, "Marketing")

ft1 = FulltimeEmployee("David", "Wilson", 70000, "Finance", "Health Insurance")
ft2 = FulltimeEmployee("Eva", "Johnson", 80000, "IT", "Retirement Plan")
```

```
print(e1.display_info())
print(e2.display_info())
print(e3.display_info())
print(ft1.display_info())
print(ft2.display_info())

print("Average Salary of all Employees:", Employee.average_salary())
```

```
→ Name: Alice, Family: Smith, Salary: 50000, Department: HR
Name: Bob, Family: Jones, Salary: 60000, Department: Engineering
Name: Charlie, Family: Brown, Salary: 55000, Department: Marketing
Name: David, Family: Wilson, Salary: 70000, Department: Finance, Benefits: Health Insurance
Name: Eva, Family: Johnson, Salary: 80000, Department: IT, Benefits: Retirement Plan
Average Salary of all Employees: 63000.0
```