Name: Rohini Patturaja

Subject: Big Data Analytics

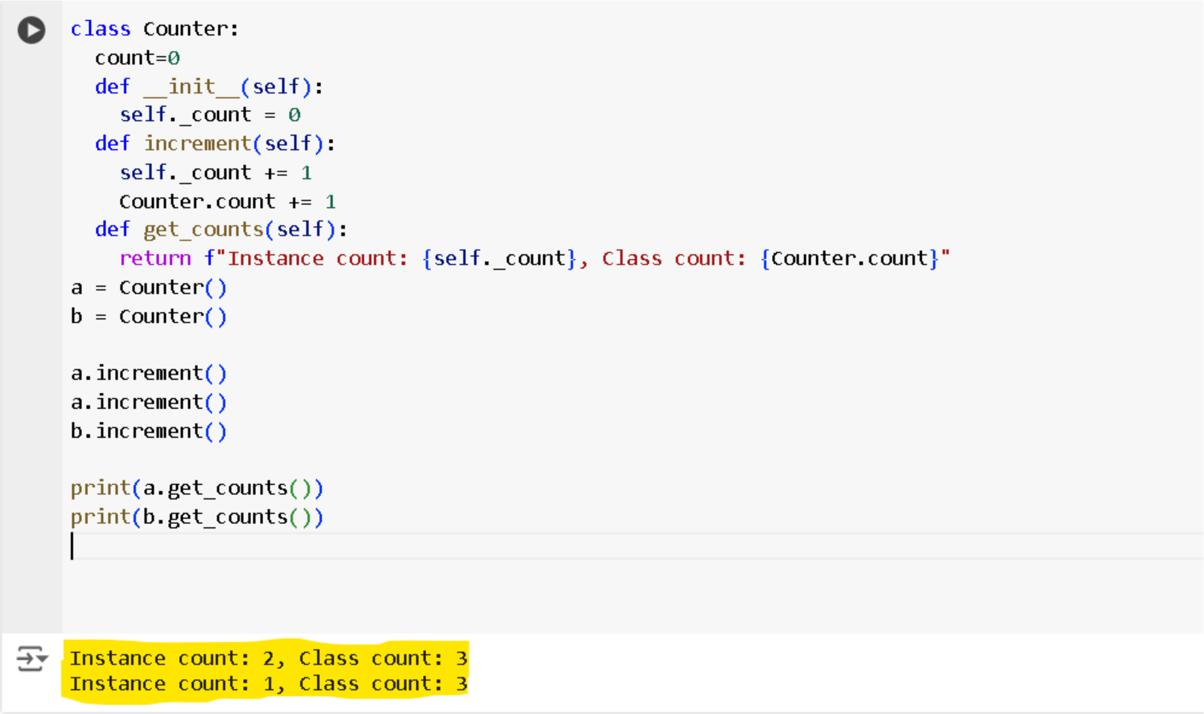
Course ID: DSA 5620

Email: rxp36770@ucmo.edu

Github link: Github ICP2 link

Video link: Rohini_Patturaja_ICP2.mp4

1. Question: Consider the following Python code:



```
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}"
a = Counter()
b = Counter()

a.increment()
a.increment()
b.increment()

print(a.get_counts())
print(b.get_counts())
```

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

Tasks:

1.Explain the difference between Counter.count and self._count?

Counter.count is a class variable, which means it is shared among all instances of a class. It is initialized to 0 and incremented every time a new instance of Counter is created.

self._count is an instance variable, which means it is unique to each instance of a class. It is initialized to 0 in the **__init__** method and incremented by the increment method.

2.What is the output of a.get_counts() and b.get_counts()?

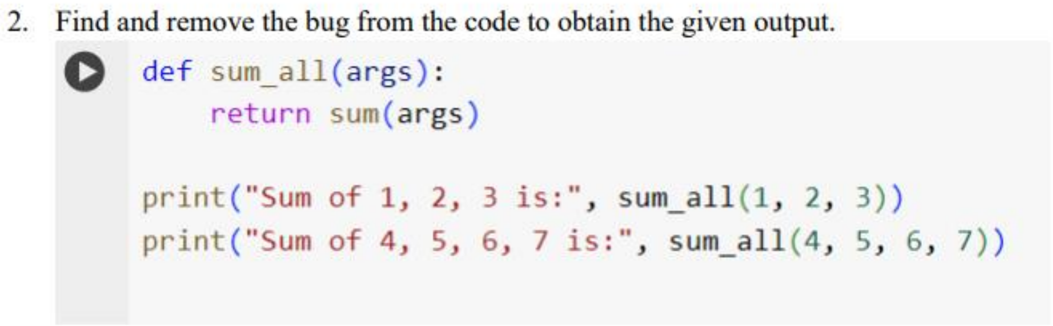
Instance count: 2, Class count: 3

Instance count: 1, Class count: 3

3.How does the increment method affect both the class and instance variables?

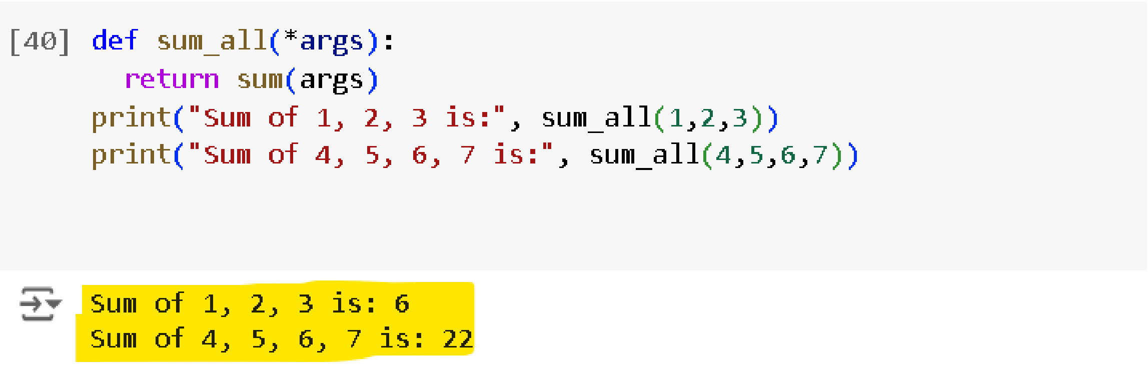
Instance Variable (_count): Each instance (a and b) has its own _count. The _count for **a** increases with each call to **a.increment(),** while _count for **b** increases with each call to **b.increment().** Therefore, the instance count for **a is 2** and for **b is 1**.

Class Variable (count): The class variable count is shared among all instances. It is incremented every time **increment** is called on any instance. Therefore, after three increments, **Counter.count is 3**.



2. Find and remove the bug from the code to obtain the given output.

```
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))
```



```
[40] 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))
```

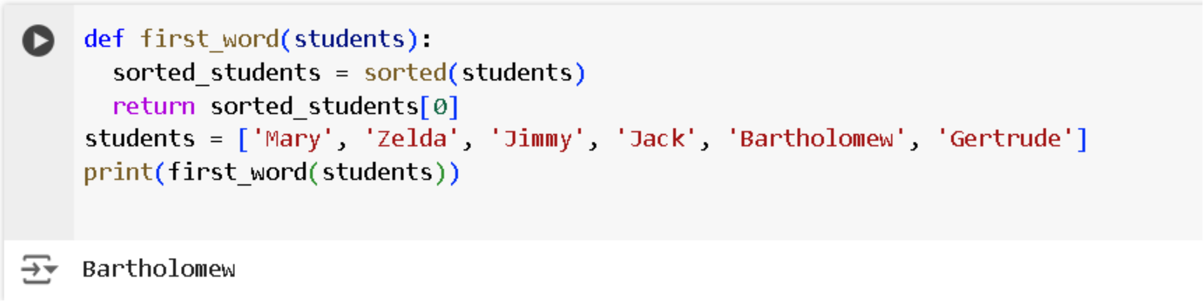
```
Sum of 1, 2, 3 is: 6
Sum of 4, 5, 6, 7 is: 22
```

3. Write a function called first_word that takes a list of character strings as input and returns the first element of the list in alphabetical order. For example, your function should work like this:

students = ['Mary', 'Zelda', 'Jimmy', 'Jack', 'Bartholomew', 'Gertrude'] (Input)

first_word(students) (Function)

'Bartholomew' (Output)

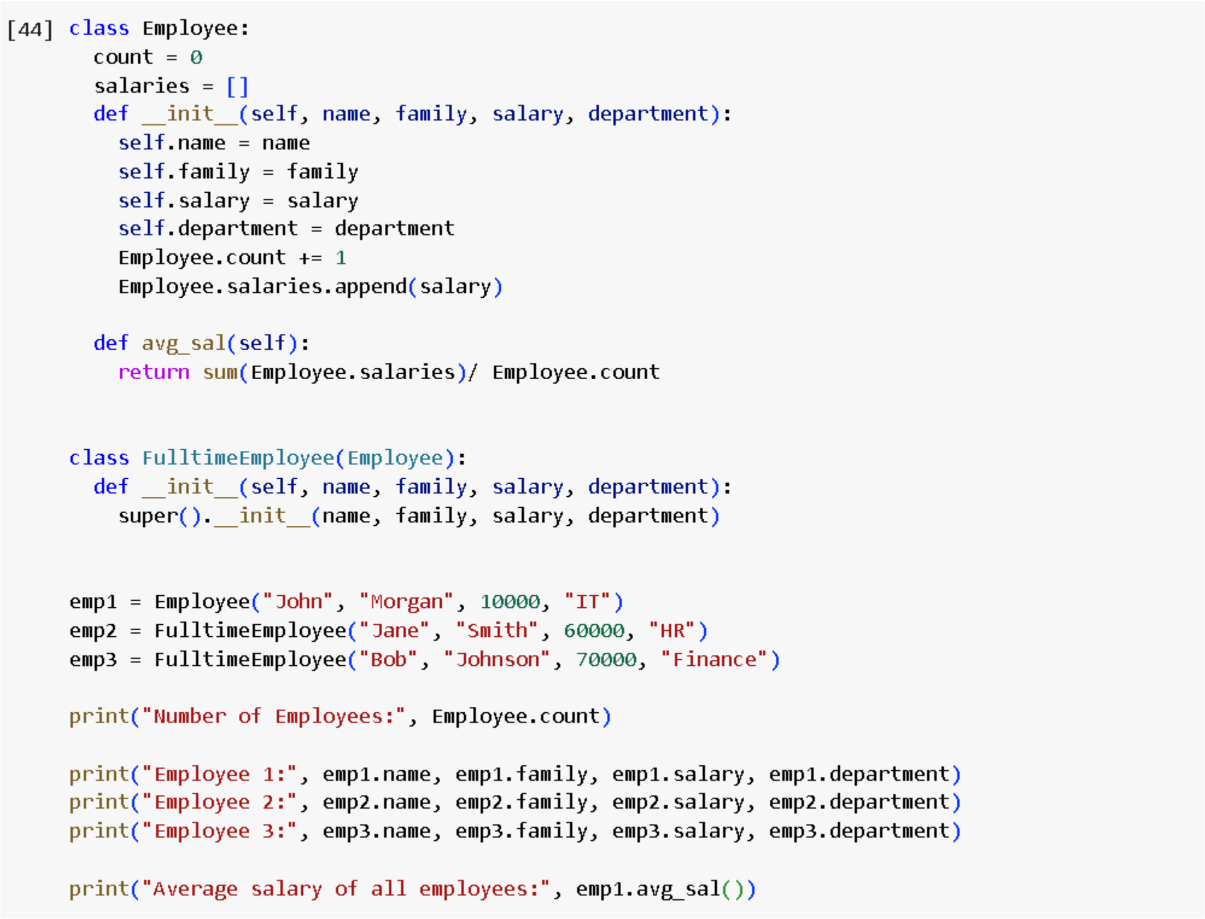


```
def first_word(students):
  sorted_students = sorted(students)
  return sorted_students[0]
students = ['Mary', 'Zelda', 'Jimmy', 'Jack', 'Bartholomew', 'Gertrude']
print(first_word(students))
```

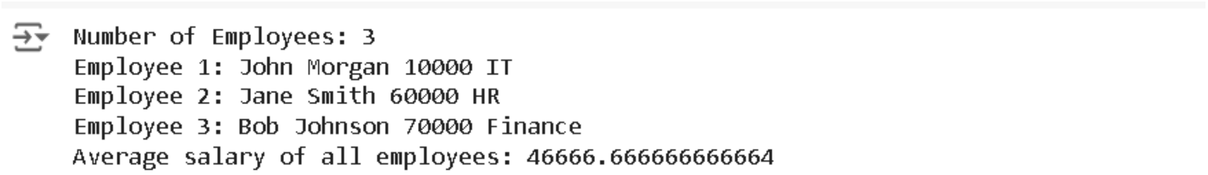
```
Bartholomew
```

4. Create a class Employee and then do the following

- Create a data member to count the number of Employees
- Create a constructor to initialize name, family, salary, department
- Create a function to average salary
- Create a Fulltime Employee class and it should inherit the properties of Employee class
- Create the instances of Fulltime Employee class and Employee class and call their member functions.



```
[44] class Employee:
       count = 0
       salaries = []
       def __init__(self, name, family, salary, department):
         self.name = name
         self.family = family
         self.salary = salary
         self.department = department
         Employee.count += 1
         Employee.salaries.append(salary)

       def avg_sal(self):
         return sum(Employee.salaries)/ Employee.count


     class FulltimeEmployee(Employee):
       def __init__(self, name, family, salary, department):
         super().__init__(name, family, salary, department)


     emp1 = Employee("John", "Morgan", 10000, "IT")
     emp2 = FulltimeEmployee("Jane", "Smith", 60000, "HR")
     emp3 = FulltimeEmployee("Bob", "Johnson", 70000, "Finance")

     print("Number of Employees:", Employee.count)

     print("Employee 1:", emp1.name, emp1.family, emp1.salary, emp1.department)
     print("Employee 2:", emp2.name, emp2.family, emp2.salary, emp2.department)
     print("Employee 3:", emp3.name, emp3.family, emp3.salary, emp3.department)

     print("Average salary of all employees:", emp1.avg_sal())
```



```
Number of Employees: 3
Employee 1: John Morgan 10000 IT
Employee 2: Jane Smith 60000 HR
Employee 3: Bob Johnson 70000 Finance
Average salary of all employees: 46666.666666666664
```