Name: Muhammad Abdullah

Roll no: 016

Section: 3A

Department: BSDSM

Task #3: Artificial Intelligence Lab

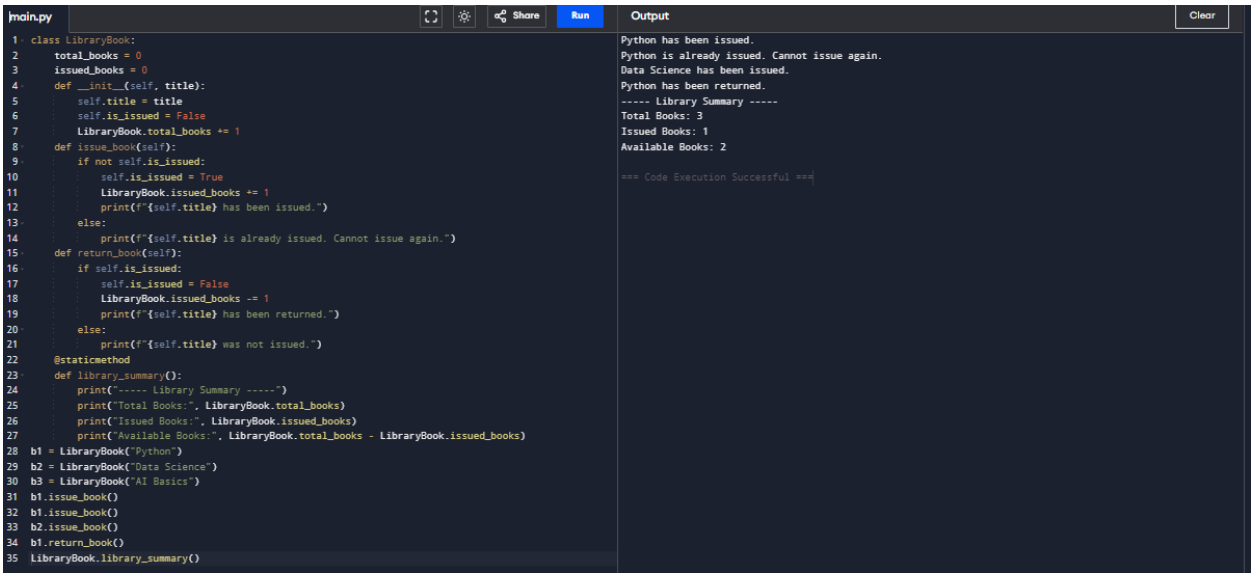
Q1. Create a payroll system using static variables to track total employees and total salary expense. Every time a new employee is added, both values must update automatically, and a static method should display the company payroll summary.

main.py | Share | Run | Output | Clear

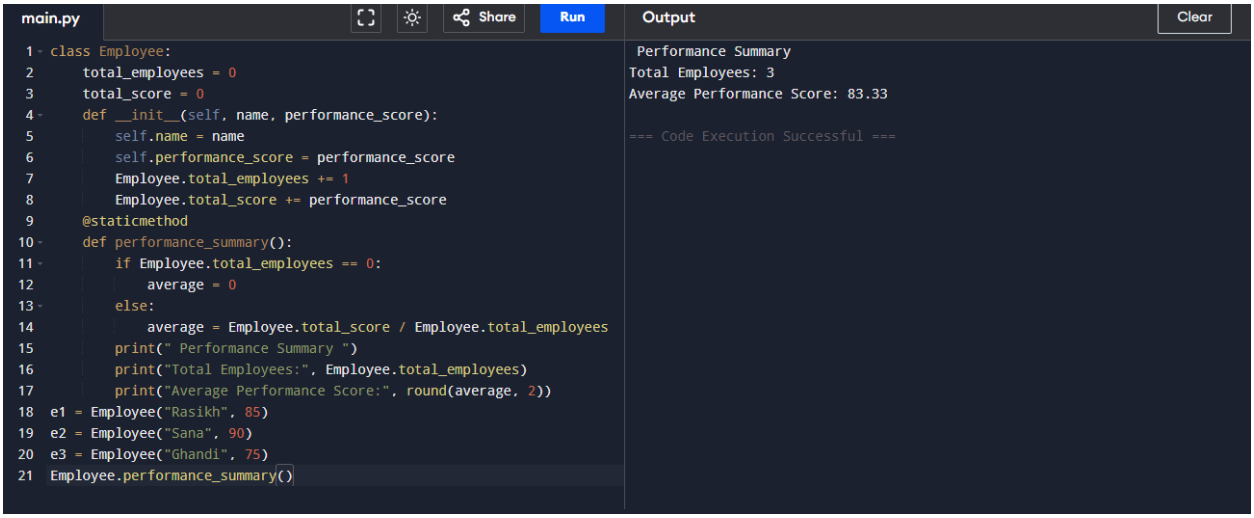
```
class Employee:
    total_employees = 0
    total_salary_expense = 0

    def __init__(self, name, salary):
        self.name = name
        self.salary = salary
        Employee.total_employees += 1
        Employee.total_salary_expense += salary

    @staticmethod
    def payroll_summary():
        print("Company Payroll Summary")
        print("Total Employees:", Employee.total_employees)
        print("Total Salary Expense:", Employee.total_salary_expense)

e1 = Employee("Rasikh", 50000)
e2 = Employee("Ali", 60000)
e3 = Employee("Jinnah", 55000)
Employee.payroll_summary()
```

```
Company Payroll Summary
Total Employees: 3
Total Salary Expense: 165000

=== Code Execution Successful ===
```

Q2. Design a library system where static variables track total books and issued books. Implement issue and return logic so that counters remain consistent and incorrect issuing is prevented.



Q3. Build an employee performance system where static data maintains total employees and average performance score, updating automatically whenever a new employee is added.



Q4. Create a class that uses a loop to process student attendance and static variables to track total present and absent students. Use if/else logic and a static method to update and display results.

main.py

```python
class Attendance:
    total_present = 0
    total_absent = 0
    def __init__(self, name):
        self.name = name
        self.is_present = False

    def mark_attendance(self, status):
        if status.lower() == "p":
            self.is_present = True
            Attendance.total_present += 1
        else:
            self.is_present = False
            Attendance.total_absent += 1
    @staticmethod
    def attendance_summary():
        print("Attendance Summary:")
        print("Total Present Students:", Attendance.total_present)
        print("Total Absent Students:", Attendance.total_absent)
students = ["Rasikh", "Sania", "Jeffery", "Donald", "Gates"]
for student_name in students:
    student = Attendance(student_name)
    status = input(f"Is {student_name} present or absent? ")
    student.mark_attendance(status)
Attendance.attendance_summary()
```

Output

```
Is Rasikh present or absent? p
Is Sania present or absent? p
Is Jeffery present or absent? p
Is Donald present or absent? a
Is Gates present or absent? a
Attendance Summary:
Total Present Students: 3
Total Absent Students: 2

=== Code Execution Successful ===
```

Q5. Design an inventory system where static variables track total items and sold items. Use loops to process sales data and if/else to prevent over-selling. Include a class method to reset inventory data

main.py

```python
class InventoryItem:
    total_items = 0
    sold_items = 0
    def __init__(self, name, quantity):
        self.name = name
        self.quantity = quantity
        InventoryItem.total_items += quantity
    def sell(self, quantity):
        if quantity <= self.quantity:
            self.quantity -= quantity
            InventoryItem.sold_items += quantity
            print(f"{quantity} {self.name}(s) sold.")
        else:
            print(f"Cannot sell {quantity} {self.name}(s). Only {self.quantity} available.")
    @classmethod
    def reset_inventory(cls):
        cls.total_items = 0
        cls.sold_items = 0
        print("Inventory data has been reset.")
    @staticmethod
    def inventory_summary():
        print("----- Inventory Summary -----")
        print("Total Items in Stock:", InventoryItem.total_items - InventoryItem.sold_items)
        print("Total Sold Items:", InventoryItem.sold_items)
item1 = InventoryItem("Laptop", 10)
item2 = InventoryItem("Phone", 20)
sales_data = [
    ("Laptop", 3),
    ("Phone", 5),
    ("Laptop", 8),
    ("Phone", 10)
]
for item_name, qty in sales_data:
    if item_name == "Laptop":
        item1.sell(qty)
    elif item_name == "Phone":
        item2.sell(qty)
InventoryItem.inventory_summary()
InventoryItem.reset_inventory()
InventoryItem.inventory_summary()
```

Output

```
3 Laptop(s) sold.
5 Phone(s) sold.
Cannot sell 8 Laptop(s). Only 7 available.
10 Phone(s) sold.
----- Inventory Summary -----
Total Items in Stock: 12
Total Sold Items: 18
Inventory data has been reset.
----- Inventory Summary -----
Total Items in Stock: 0
Total Sold Items: 0

=== Code Execution Successful ===
```