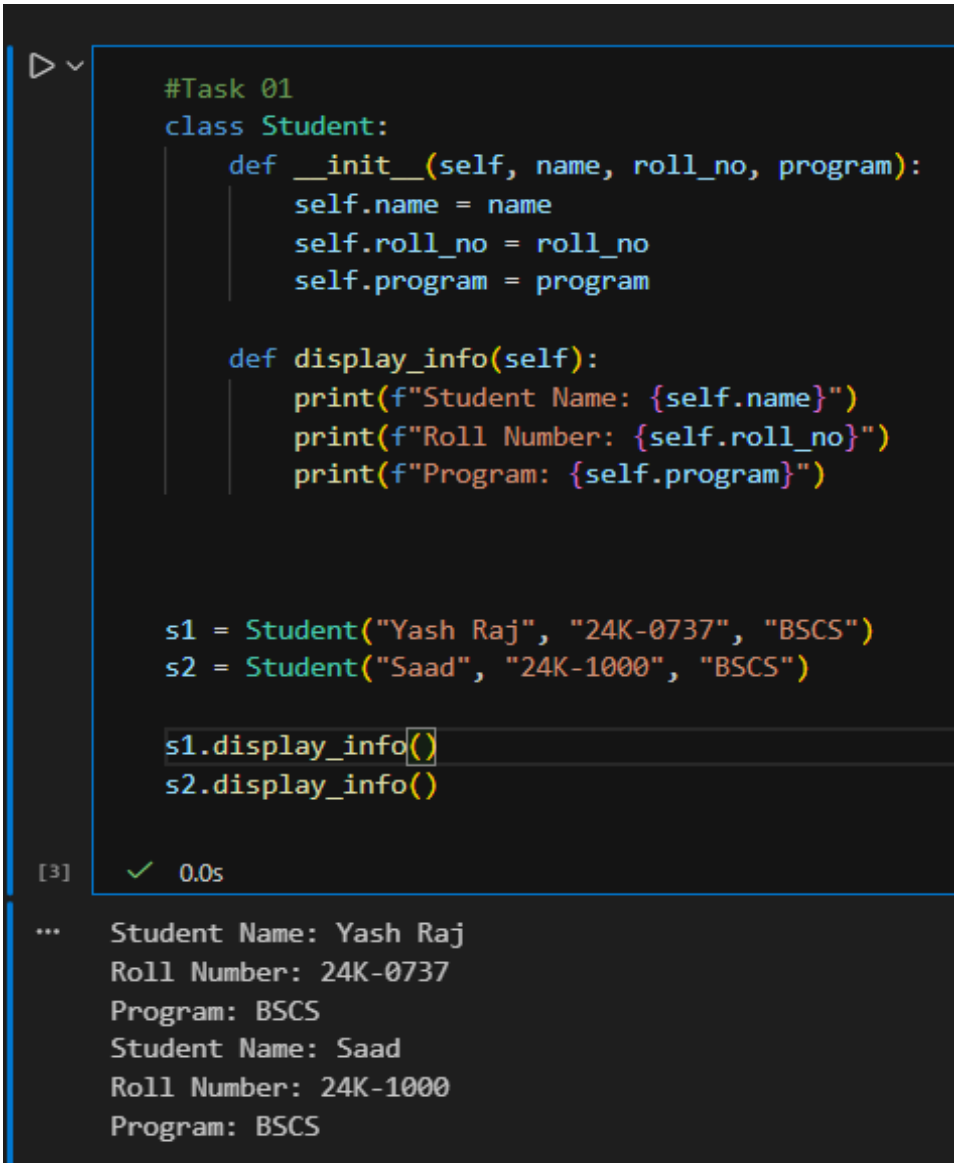
AI-Lab02 Tasks

Name: Yash Raj

Roll-Number : 24k-0737

Task 01:

Code & Output:



```python
#Task 01
class Student:
    def __init__(self, name, roll_no, program):
        self.name = name
        self.roll_no = roll_no
        self.program = program

    def display_info(self):
        print(f"Student Name: {self.name}")
        print(f"Roll Number: {self.roll_no}")
        print(f"Program: {self.program}")



s1 = Student("Yash Raj", "24K-0737", "BSCS")
s2 = Student("Saad", "24K-1000", "BSCS")

s1.display_info()
s2.display_info()
```

[3] ✓ 0.0s

```
Student Name: Yash Raj
Roll Number: 24K-0737
Program: BSCS
Student Name: Saad
Roll Number: 24K-1000
Program: BSCS
```

Task 02:

Code & Output:

```python
#Task 02
class SmartFan:
    def __init__(self, room_name, speed=0):
        self.room_name = room_name
        self.speed = speed

    def speed_up(self):
        self.speed += 1

    def slow_down(self):
        if self.speed > 0:
            self.speed -= 1

    def show_details(self):
        print(f"{self.room_name} fan speed is {self.speed}")


fan_a = SmartFan("Bedroom")
fan_b = SmartFan("Living Room")

fan_a.speed_up()
fan_a.speed_up()
fan_a.show_details()

fan_b.speed_up()
fan_b.show_details()
```

[4] ✓ 0.0s

```
Bedroom fan speed is 2
Living Room fan speed is 1
```

Task 03:

Code & Output:

```python
#Task 03
class Book:
    library_name = "Fast Library"

    def __init__(self, title, author):
        self.title = title
        self.author = author

    def book_details(self):
        print(f"Title: {self.title}")
        print(f"Author: {self.author}")
        print(f"Library: {Book.library_name}")


b1 = Book("Python 101", "Mark")
b2 = Book("AI Basics", "Andrew Ng")

b1.book_details()
b2.book_details()

Book.library_name = "University Library"

b1.book_details()
b2.book_details()
```

[6] ✓ 0.0s

```
Title: Python 101
Author: Mark
Library: Fast Library
Title: AI Basics
Author: Andrew Ng
Library: Fast Library
Title: Python 101
Author: Mark
Library: University Library
Title: AI Basics
Author: Andrew Ng
Library: University Library
```

Task 04:

```python
#Task 04
class BankAccount:
    def __init__(self, holder, number, balance=0):
        self.holder = holder
        self.number = number
        self.balance = balance

    def add_money(self, amount):
        self.balance += amount

    def take_money(self, amount):
        if amount <= self.balance:
            self.balance -= amount
        else:
            print("Not enough balance")

    def account_summary(self):
        print(f"Account Holder: {self.holder}")
        print(f"Account No: {self.number}")
        print(f"Balance: {self.balance}")



acc1 = BankAccount("Yash Raj", "5001", 40001010)
acc2 = BankAccount("Jiyanshu", "5002", 600020002)

acc1.add_money(1000)
acc1.take_money(1500)
acc1.account_summary()

acc2.take_money(2000)
acc2.account_summary()
```

✓ 0.0s

```
Account Holder: Yash Raj
Account No: 5001
Balance: 40000510
Account Holder: Jiyanshu
Account No: 5002
Balance: 600018002
```

Task 05:

```
#Task 05
class Light:
    def __init__(self , room_name, status = False):
        self.room_name = room_name
        self.status = status
    def Turn_on(self):
        self.status = True
        print(f"{self.room_name} Light turned On")
    def Turn_off(self):
        self.status = False
        print(f"{self.room_name} Light turned off")
    def Display_info(self):
        if self.status:
            print(f"{self.room_name} light is On")
        else:
            print(f"{self.room_name} light is off")
LivingRoom = Light("Living Room")
Bedroom = Light("Bedroom")
LivingRoom.Turn_on()
LivingRoom.Display_info()
Bedroom.Turn_on()
Bedroom.Display_info()

print("After Turning off : ")
Bedroom.Turn_off()
LivingRoom.Turn_off()
LivingRoom.Display_info()
Bedroom.Display_info()
```

[10] ✓ 0.0s

```
Living Room Light turned On
Living Room light is On
Bedroom Light turned On
Bedroom light is On
After Turning off :
Bedroom Light turned off
Living Room Light turned off
Living Room light is off
Bedroom light is off
```

Task 06 :

```python
#Task 06
class Staff:
    def __init__(self, name, staff_id, department):
        self.name = name
        self.staff_id = staff_id
        self.department = department

    def display_info(self):
        print(self.name, self.staff_id, self.department)


class Teacher(Staff):
    def __init__(self, name, staff_id, department, courses, salary):
        super().__init__(name, staff_id, department)
        self.courses = courses
        self.salary = salary

    def display_info(self):
        print("Teacher:", self.name)
        print("Courses:", self.courses)
        print("Salary:", self.salary)


class AdminStaff(Staff):
    def __init__(self, name, staff_id, department, role, hours):
        super().__init__(name, staff_id, department)
        self.role = role
        self.hours = hours


class ResearchAssistant(Staff):
    def __init__(self, name, staff_id, department, topic, stipend):
        super().__init__(name, staff_id, department)
        self.topic = topic
        self.stipend = stipend


t = Teacher("Sir Riaz", "T10", "CS", ["AI", "ML"], 180000)
a = AdminStaff("Sir Abdullah", "D23", "Admin", "Officer", 8)
r = ResearchAssistant("Saira", "R10", "CS", "Robotics", 55000)

t.display_info()
a.display_info()
r.display_info()
```

[11] ✓ 0.0s

```
Teacher: Sir Riaz
Courses: ['AI', 'ML']
Salary: 180000
Sir Abdullah D23 Admin
Saira R10 CS
```

Task 07:

```
#Task 07
class Vehicle:
    def start(self):
        print("Vehicle is starting")


class Car(Vehicle):
    def start(self):
        print("Car has started")


class Bike(Vehicle):
    def start(self):
        print("Bike has started")


class Bus(Vehicle):
    def start(self):
        print("Bus has started")


v_list = [Car(), Bike(), Bus()]

for v in v_list:
    v.start()
```

[12] ✓ 0.0s

```
Car has started
Bike has started
Bus has started
```

Task 08:

```
#Task 08
class Calculator:
    def multiply_values(self, *nums):
        result = 1
        for n in nums:
            result *= n
        return result


calc = Calculator()
print(calc.multiply_values(2, 3))
print(calc.multiply_values(2, 3, 4))
print(calc.multiply_values(1, 2, 3, 4, 5))
```

[13] ✓ 0.0s

```
6
24
120
```

Task 09:

```python
#Task 09
class Employee:
    def __init__(self, name, salary):
        self._name = name
        self._salary = salary


class Manager(Employee):
    def Display_Employee(self):
        print(f"Name: {self._name}")
        print(f"Salary: {self._salary}")


m = Manager("Yash Raj", 130000)
m.Display_Employee()
```

[14] ✓ 0.0s

```
Name: Yash Raj
Salary: 130000
```

Task 10:

```python
#Task 10
class BankAccount:
    def __init__(self, balance):
        self.__balance = balance

    def deposit(self, amount):
        self.__balance += amount

    def withdraw(self, amount):
        if amount <= self.__balance:
            self.__balance -= amount
        else:
            print("Insufficient balance")

    def get_balance(self):
        return self.__balance


acc = BankAccount(7000)
acc.deposit(2000)
acc.withdraw(1000)
print("Current Balance:", acc.get_balance())
```

[15] ✓ 0.0s

```
Current Balance: 8000
```