

Name: Bisma Farhat

Sap id:39967

Task1

+ Code + Text

✓  
0s

▶

```
def sum_list(numbers):  
    total = sum(numbers)  
    return total  
  
# Sample list  
sample_list = [7, 5, 3, 0, 2]  
result = sum_list(sample_list)  
print("The sum is:", result)
```

↔

The sum is: 17

✓  
0s


[3] Start coding or generate with AI.


## Task2

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

✓  
0s

 `def check_attendance(roll_number, attendance_list):`  
  
    `if roll_number in attendance_list:`  
        `return "Present"`  
    `else:`  
        `return "Absent"`  
  
    `attendance_list = [101, 102, 103, 104, 105]`  
    `roll_number_to_check = 102`  
  
    `result = check_attendance(roll_number_to_check, attendance_list)`  
    `print(f"Roll number {roll_number_to_check} is {result}.")`

 Roll number 102 is Present.

✓  
0s

[3] Start coding or generate with AI.

## Task3

```
class Student:
    def __init__(self, name, roll_number, grade):
        self.name = name
        self.roll_number = roll_number
        self.grade = grade

    def display_info(self):
        """Method to display student information."""
        print(f"Name: {self.name}, Roll Number: {self.roll_number}, Grade: {self.grade}")

student1 = Student("Alice", 101, "A")

print("Initial Student Information:")
student1.display_info()

student1.name = "Alice Smith"
student1.roll_number = 102
student1.grade = "A+"
```



Untitled.ipynb

File Edit View Insert Runtime Tools Help [All changes saved](#)

+ Code + Text

✓  
0s



```
student1.roll_number = 102
student1.grade = "A+"
```

```
print("\nUpdated Student Information:")
student1.display_info()
```



```
Initial Student Information:
Name: Alice, Roll Number: 101, Grade: A

Updated Student Information:
Name: Alice Smith, Roll Number: 102, Grade: A+
```

✓  
0s

[3] Start coding or [generate](#) with AI.

## Task4

✓  
0s

▶

+ Code

+ Text

```
class Student:
    def __init__(self, name, roll_number, grade):
        self.name = name
        self.roll_number = roll_number
        self.grade = grade

    def display_info(self):
        """Method to display student information."""
        print(f"Name: {self.name}, Roll Number: {self.roll_number}, Grade: {self.grade}")

student1 = Student("Alice", 101, "A")
student2 = Student("Bob", 102, "B")
student3 = Student("Charlie", 103, "A+")

print("Student Information:")
student1.display_info()
student2.display_info()
student3.display_info()

print("\nUpdated Student Information:")
```



File Edit View Insert Runtime Tools Help All changes saved



+ Code + Text



0s

```
student3.display_info()
```

```
print("\nUpdated Student Information:")
student1.display_info()
```



Student Information:

Name: Alice, Roll Number: 101, Grade: A

Name: Bob, Roll Number: 102, Grade: B

Name: Charlie, Roll Number: 103, Grade: A+

Updated Student Information:

Name: Alice, Roll Number: 101, Grade: A



0s

[3] Start coding or generate with AI.



## Task5

+ Code

+ Text

```
class Student:
    def __init__(self, name, age, grades):
        self.name = name
        self.age = age
        self.grades = grades

    def average_grade(self):
        if not self.grades:
            return 0
        return sum(self.grades) / len(self.grades)

student = Student("Alice", 20, [85, 90, 78, 92])
print(f"Average grade for {student.name}: {student.average_grade()}")
```

↔

 Average grade for Alice: 86.25

✓  
0s [3] Start coding or [generate](#) with AI.