

IN LAB

LAB TASK 1:

```
"""
Created on Thu Oct 5 15:19:13 2023

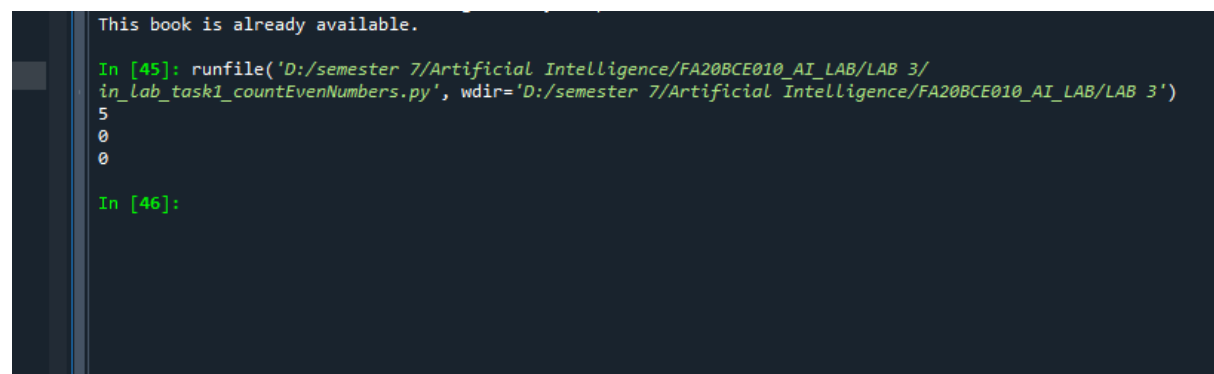
@author: MINUSX
"""

def count_even_numbers(number_list):
    even_count = 0
    for num in number_list:
        if num % 2 == 0:
            even_count+=1

    return even_count

numbers1 = [1,2,3,4,5,6,7,8,9,10]
numbers2 = [11,13,15,17]
numbers3 = []

print(count_even_numbers(numbers1))
print(count_even_numbers(numbers2))
print(count_even_numbers(numbers3))
```



```
This book is already available.

In [45]: runfile('D:/semester 7/Artificial Intelligence/FA20BCE010_AI_LAB/LAB 3/
in_lab_task1_countEvenNumbers.py', wdir='D:/semester 7/Artificial Intelligence/FA20BCE010_AI_LAB/LAB 3')
5
0
0

In [46]:
```

LAB TASK 2:

```
"""
Created on Thu Oct 5 15:33:26 2023

@author: MINUSX
"""
```

```
def calculate_gpa(students):  
  
    #created a list to append data in result  
    results= []  
  
    for student in students:  
        name = student['name']  
        marks = student['marks']  
        total_marks = sum(marks)  
        percentage = total_marks/len(marks)  
  
        if percentage >=85:  
            grade = 'A'  
            grade_points = 4.00  
  
        elif percentage >= 80:  
            grade = 'A-'  
            grade_points = 3.66  
  
        elif percentage >= 75:  
            grade = 'B+'  
            grade_points = 3.33  
  
        elif percentage >= 71:  
            grade = 'B'  
            grade_points = 3.00  
  
        elif percentage >= 68:  
            grade = 'B-'  
            grade_points = 2.66  
  
        elif percentage >= 64:  
            grade = 'C+'  
            grade_points = 2.33  
  
        elif percentage >= 61:  
            grade = 'C'  
            grade_points = 2.00  
  
        elif percentage >= 58:  
            grade = 'C-'  
            grade_points = 1.66  
  
        elif percentage >= 54:  
            grade = 'D+'  
            grade_points = 1.30  
  
        elif percentage >= 50:  
            grade = 'D'  
            grade_points = 1.00  
  
        else:  
            grade = 'F'  
            grade_points = 0.00
```

```

        results.append({
            'name': name,
            'grade': [grade ],
            'grade_points': [grade_points for _ in marks],
            'gpa': round(grade_points, 2)
        })

    return results

students_data = [
    {'name': 'bilal', 'marks': [78, 85, 92, 69, 77]},
    {'name': 'abdullah', 'marks': [90, 88, 75, 81, 95]},
    {'name': 'zain', 'marks': [63, 76, 58, 70, 72]},
    {'name': 'umar', 'marks': [48, 53, 60, 42, 55]},
    {'name': 'ali', 'marks': [76, 79, 81, 82, 77]}
]

# Calculate GPA for each student
gpa_results = calculate_gpa(students_data)

# Print the results
for result in gpa_results:
    print(result)

```

```

{ 'name': 'ali', 'grade': ['B+'], 'grade_points': [3.33, 3.33, 3.33, 3.33, 3.33], 'gpa': 3.33}

In [47]: runfile('D:/semester 7/Artificial Intelligence/FA20BCE010_AI_LAB/LAB 3/in_lab_task2_calculateGpa.py',
wdir='D:/semester 7/Artificial Intelligence/FA20BCE010_AI_LAB/LAB 3')
{'name': 'bilal', 'grade': ['A-'], 'grade_points': [3.66, 3.66, 3.66, 3.66, 3.66], 'gpa': 3.66}
{'name': 'abdullah', 'grade': ['A'], 'grade_points': [4.0, 4.0, 4.0, 4.0, 4.0], 'gpa': 4.0}
{'name': 'zain', 'grade': ['C+'], 'grade_points': [2.33, 2.33, 2.33, 2.33, 2.33], 'gpa': 2.33}
{'name': 'umar', 'grade': ['D'], 'grade_points': [1.0, 1.0, 1.0, 1.0, 1.0], 'gpa': 1.0}
{'name': 'ali', 'grade': ['B+'], 'grade_points': [3.33, 3.33, 3.33, 3.33, 3.33], 'gpa': 3.33}

In [48]:

```

LAB TASK 3:

```

# -*- coding: utf-8 -*-
"""
Created on Thu Oct 5 16:57:00 2023

@author: MINUSX
"""

class Student:

```

```
def __init__(self, name, roll_number):
    self.name = name
    self.roll_number = roll_number
    self.marks = []

def add_marks(self, subject, mark):
    self.marks.append((subject, mark))

def calculate_average(self):
    if not self.marks:
        return 0

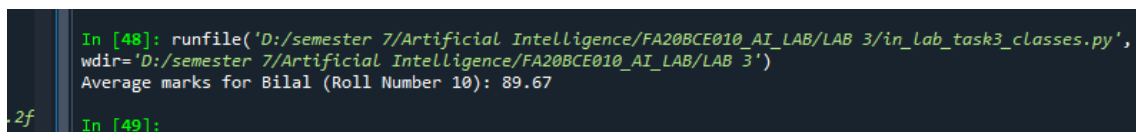
    total_marks = sum(mark for subject, mark in self.marks)
    average = total_marks / len(self.marks)
    return average

# Create an instance of the Student class
student1 = Student("Bilal", 10)

# Add some marks
student1.add_marks("Math", 96)
student1.add_marks("Science", 85)
student1.add_marks("English", 88)

# Calculate and print the average marks

average_marks = student1.calculate_average()
print(f"Average marks for {student1.name} (Roll Number {student1.roll_number}): {average_marks:.2f}")
```



```
In [48]: runfile('D:/semester 7/Artificial Intelligence/FA20BCE010_AI_LAB/LAB 3/in_lab_task3_classes.py',
wdir='D:/semester 7/Artificial Intelligence/FA20BCE010_AI_LAB/LAB 3')
Average marks for Bilal (Roll Number 10): 89.67

In [49]:
```

Post LAB

TASK

```
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author
        self.available = True

    def borrow(self):
```

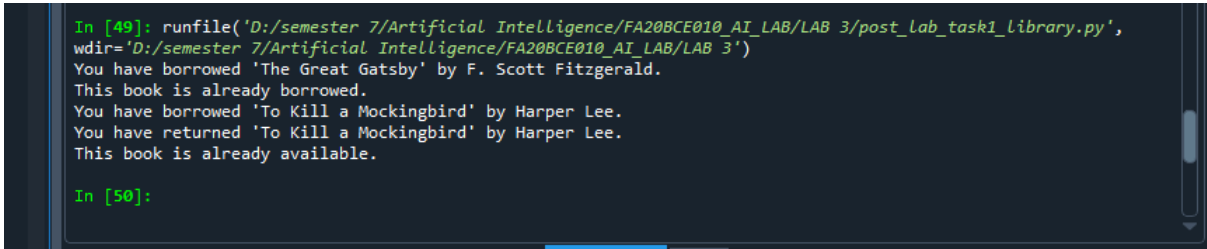
```
    if self.available:
        self.available = False
        print(f"You have borrowed '{self.title}' by {self.author}.")
    else:
        print("This book is already borrowed.")

def return_book(self):
    if not self.available:
        self.available = True
        print(f"You have returned '{self.title}' by {self.author}.")
    else:
        print("This book is already available.")

# Sample usage
if __name__ == "__main__":
    book1 = Book("The Great Gatsby", "F. Scott Fitzgerald")
    book2 = Book("To Kill a Mockingbird", "Harper Lee")

    book1.borrow()
    book1.borrow() # Trying to borrow again

    book2.borrow()
    book2.return_book()
    book2.return_book() # Trying to return again
```



```
In [49]: runfile('D:/semester 7/Artificial Intelligence/FA20BCE010_AI_LAB/LAB 3/post_lab_task1_library.py',
wdir='D:/semester 7/Artificial Intelligence/FA20BCE010_AI_LAB/LAB 3')
You have borrowed 'The Great Gatsby' by F. Scott Fitzgerald.
This book is already borrowed.
You have borrowed 'To Kill a Mockingbird' by Harper Lee.
You have returned 'To Kill a Mockingbird' by Harper Lee.
This book is already available.

In [50]:
```