# Assignment-5.1.6

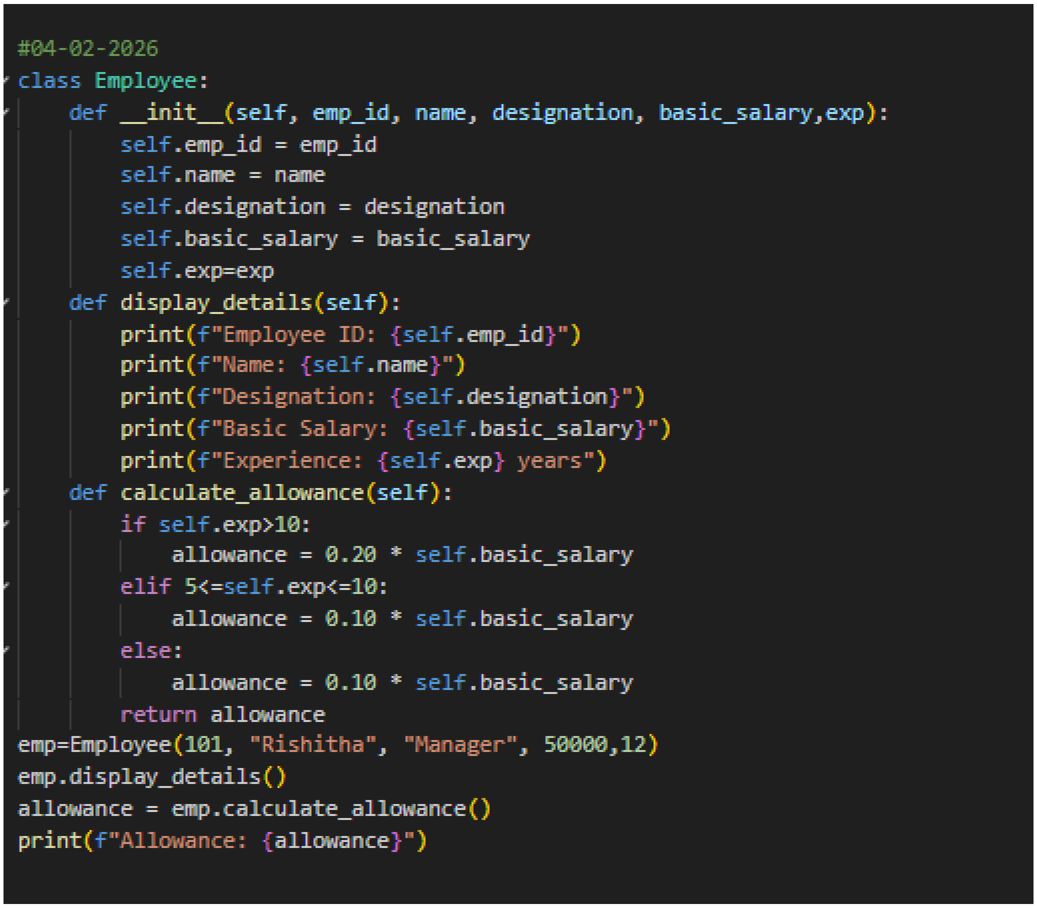
2303A51680

K Rishitha

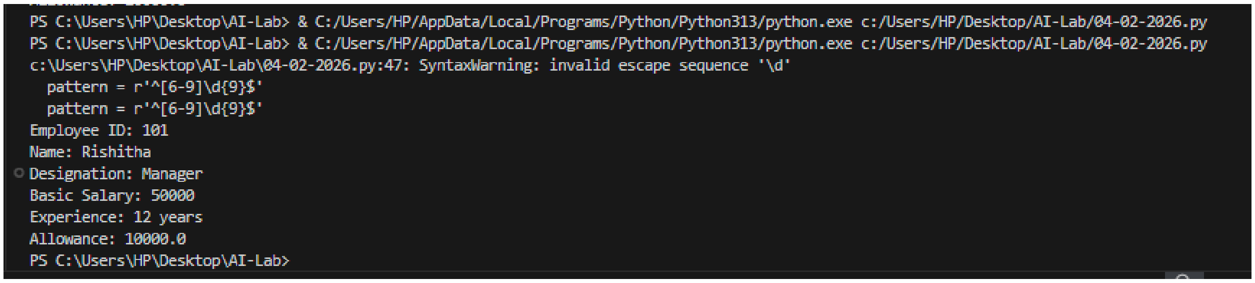
Batch-23

Task 1:
Employee Data: Create Python code that defines a class named `Employee` with the following attributes: `empid`, `empname`, `designation`, `basic_salary`, and `exp`. Implement a method `display_details()` to print all employee details. Implement another method `calculate_allowance()` to determine additional allowance based on experience:
- If `exp > 10 years` → allowance = 20% of `basic_salary`
- If `5 ≤ exp ≤ 10 years` → allowance = 10% of `basic_salary`
- If `exp < 5 years` → allowance = 5% of `basic_salary`

Finally, create at least one instance of the `Employee` class, call the `display_details()` method, and print the calculated allowance.



```python
#04-02-2026
class Employee:
    def __init__(self, emp_id, name, designation, basic_salary,exp):
        self.emp_id = emp_id
        self.name = name
        self.designation = designation
        self.basic_salary = basic_salary
        self.exp=exp
    def display_details(self):
        print(f"Employee ID: {self.emp_id}")
        print(f"Name: {self.name}")
        print(f"Designation: {self.designation}")
        print(f"Basic Salary: {self.basic_salary}")
        print(f"Experience: {self.exp} years")
    def calculate_allowance(self):
        if self.exp>10:
            allowance = 0.20 * self.basic_salary
        elif 5<=self.exp<=10:
            allowance = 0.10 * self.basic_salary
        else:
            allowance = 0.10 * self.basic_salary
        return allowance
emp=Employee(101, "Rishitha", "Manager", 50000,12)
emp.display_details()
allowance = emp.calculate_allowance()
print(f"Allowance: {allowance}")
```



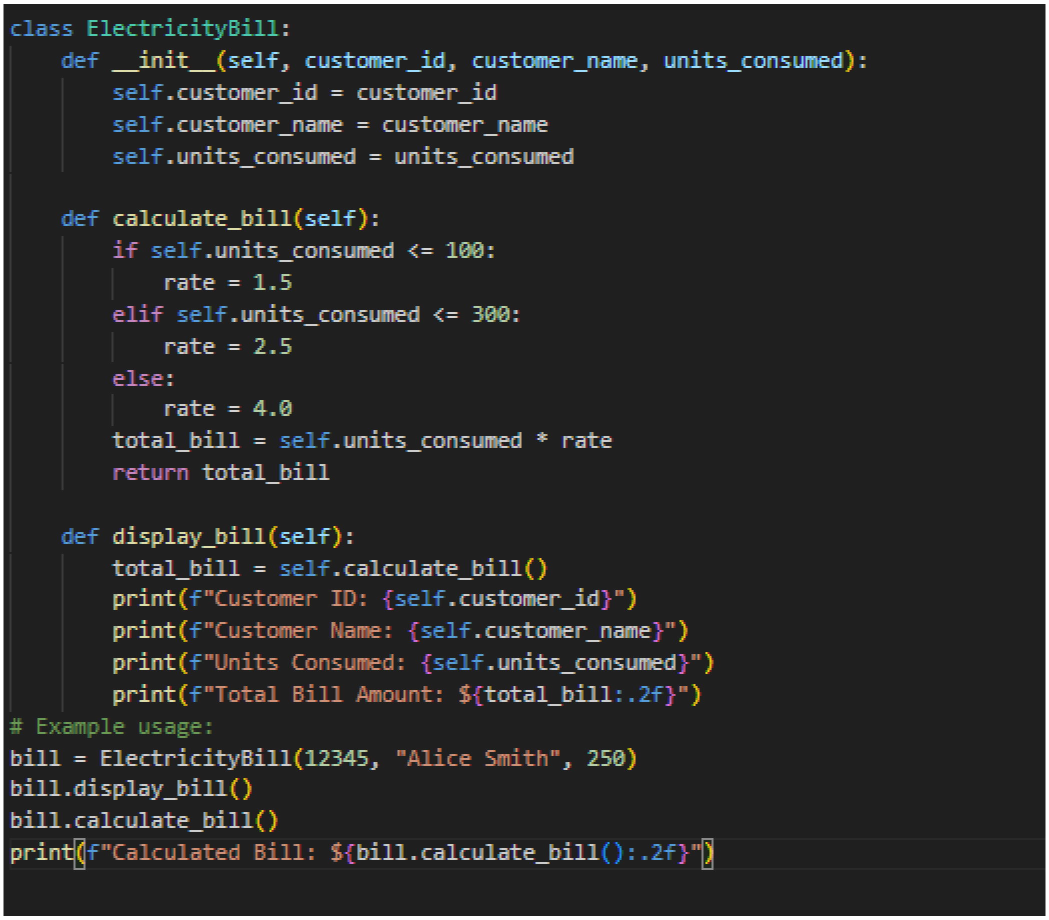
```
PS C:\Users\HP\Desktop\AI-Lab> & C:/Users/HP/AppData/Local/Programs/Python/Python313/python.exe c:/Users/HP/Desktop/AI-Lab/04-02-2026.py
PS C:\Users\HP\Desktop\AI-Lab> & C:/Users/HP/AppData/Local/Programs/Python/Python313/python.exe c:/Users/HP/Desktop/AI-Lab/04-02-2026.py
c:\Users\HP\Desktop\AI-Lab\04-02-2026.py:47: SyntaxWarning: invalid escape sequence '\d'
  pattern = r'^[6-9]\d{9}$'
  pattern = r'^[6-9]\d{9}$'
Employee ID: 101
Name: Rishitha
Designation: Manager
Basic Salary: 50000
Experience: 12 years
Allowance: 10000.0
PS C:\Users\HP\Desktop\AI-Lab>
```

Task 2:

Electricity Bill Calculation- Create Python code that defines a class named `ElectricityBill` with attributes: `customer_id`, `name`, and `units_consumed`. Implement a method `display_details()` to print customer details, and a method `calculate_bill()` where:

- Units ≤ 100 → ₹5 per unit
- 101 to 300 units → ₹7 per unit
- More than 300 units → ₹10 per unit

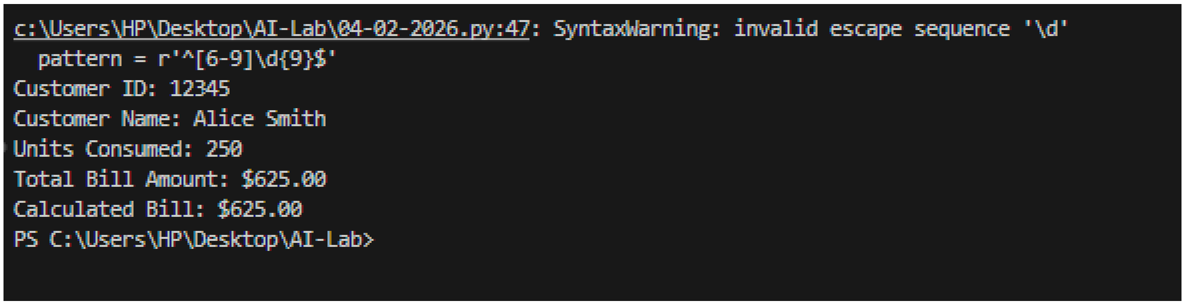
Create a bill object, display details, and print the total bill amount.



```python
class ElectricityBill:
    def __init__(self, customer_id, customer_name, units_consumed):
        self.customer_id = customer_id
        self.customer_name = customer_name
        self.units_consumed = units_consumed

    def calculate_bill(self):
        if self.units_consumed <= 100:
            rate = 1.5
        elif self.units_consumed <= 300:
            rate = 2.5
        else:
            rate = 4.0
        total_bill = self.units_consumed * rate
        return total_bill

    def display_bill(self):
        total_bill = self.calculate_bill()
        print(f"Customer ID: {self.customer_id}")
        print(f"Customer Name: {self.customer_name}")
        print(f"Units Consumed: {self.units_consumed}")
        print(f"Total Bill Amount: ${total_bill:.2f}")
# Example usage:
bill = ElectricityBill(12345, "Alice Smith", 250)
bill.display_bill()
bill.calculate_bill()
print(f"Calculated Bill: ${bill.calculate_bill():.2f}")
```



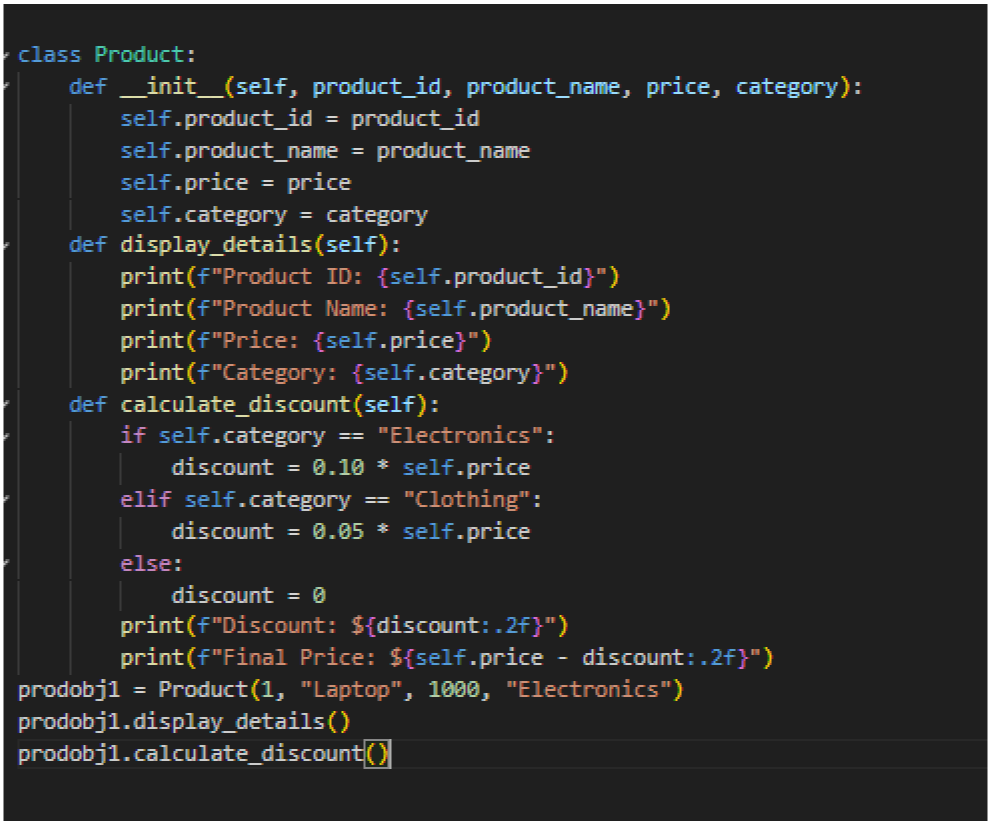
```
c:\Users\HP\Desktop\AI-Lab\04-02-2026.py:47: SyntaxWarning: invalid escape sequence '\d'
  pattern = r'^[6-9]\d{9}$'
Customer ID: 12345
Customer Name: Alice Smith
Units Consumed: 250
Total Bill Amount: $625.00
Calculated Bill: $625.00
PS C:\Users\HP\Desktop\AI-Lab>
```

Task 3:

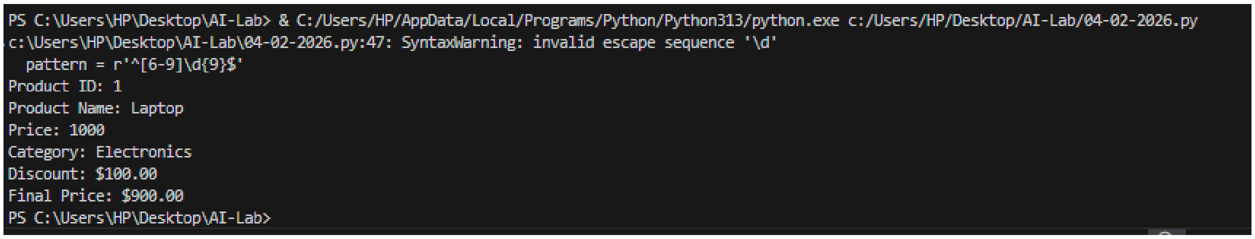
Product Discount Calculation- Create Python code that defines a class named `Product` with attributes: `product_id`, `product_name`, `price`, and `category`. Implement a method `display_details()` to print product details. Implement another method `calculate_discount()` where:

- Electronics → 10% discount
- Clothing → 15% discount
- Grocery → 5% discount

Create at least one product object, display details, and print the final price after discount.



```python
class Product:
    def __init__(self, product_id, product_name, price, category):
        self.product_id = product_id
        self.product_name = product_name
        self.price = price
        self.category = category
    def display_details(self):
        print(f"Product ID: {self.product_id}")
        print(f"Product Name: {self.product_name}")
        print(f"Price: {self.price}")
        print(f"Category: {self.category}")
    def calculate_discount(self):
        if self.category == "Electronics":
            discount = 0.10 * self.price
        elif self.category == "Clothing":
            discount = 0.05 * self.price
        else:
            discount = 0
        print(f"Discount: ${discount:.2f}")
        print(f"Final Price: ${self.price - discount:.2f}")
prodobj1 = Product(1, "Laptop", 1000, "Electronics")
prodobj1.display_details()
prodobj1.calculate_discount()
```



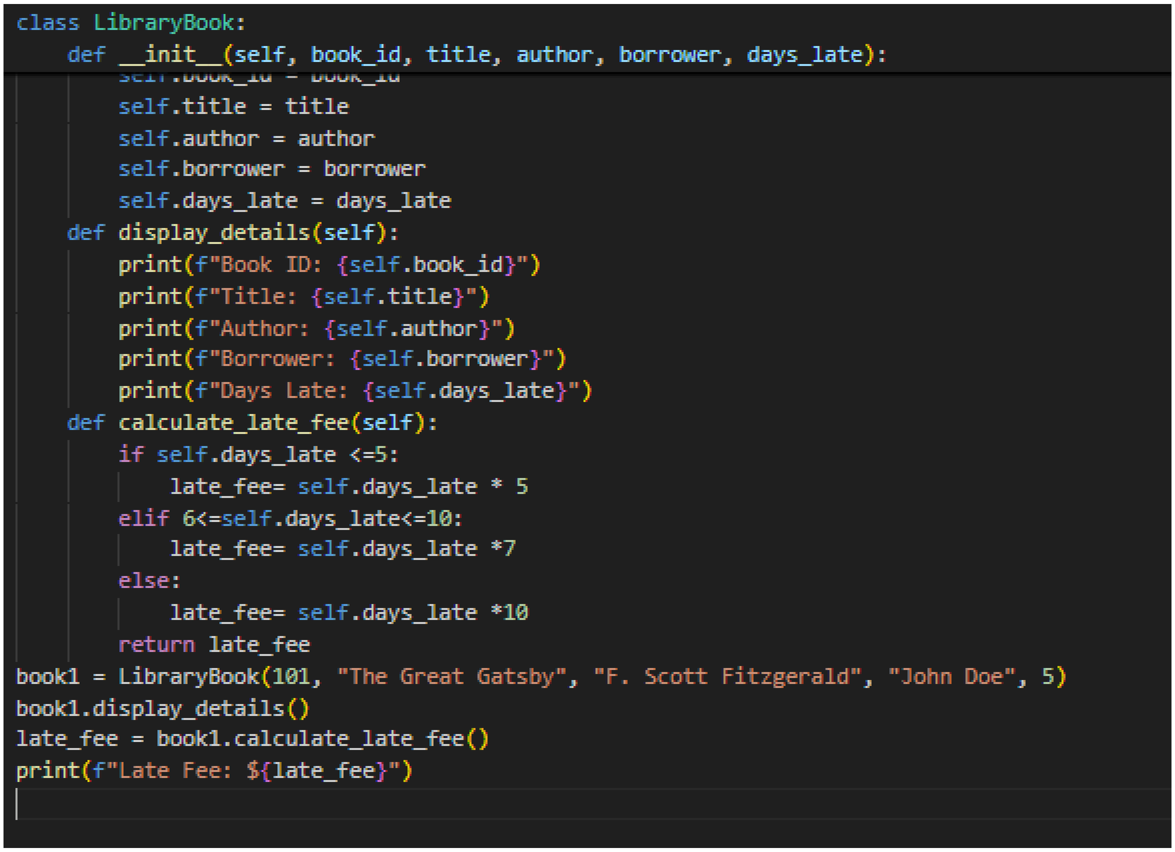
```
PS C:\Users\HP\Desktop\AI-Lab> & C:/Users/HP/AppData/Local/Programs/Python/Python313/python.exe c:/Users/HP/Desktop/AI-Lab/04-02-2026.py
c:\Users\HP\Desktop\AI-Lab\04-02-2026.py:47: SyntaxWarning: invalid escape sequence '\d'
  pattern = r'^[6-9]\d{9}$'
Product ID: 1
Product Name: Laptop
Price: 1000
Category: Electronics
Discount: $100.00
Final Price: $900.00
PS C:\Users\HP\Desktop\AI-Lab>
```

Task 4:

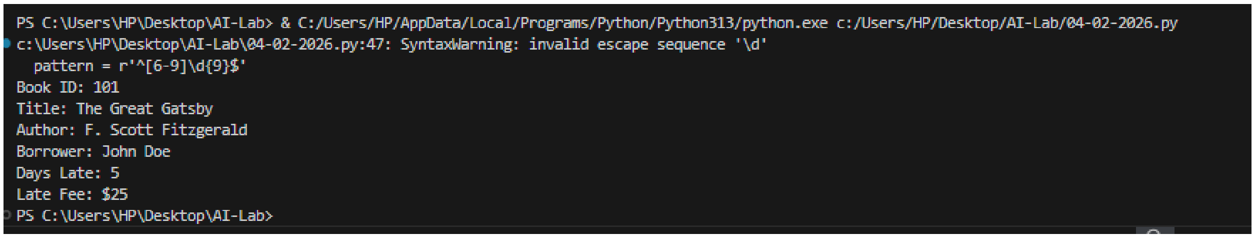
Book Late Fee Calculation- Create Python code that defines a class named `LibraryBook` with attributes: `book_id`, `title`, `author`, `borrower`, and `days_late`. Implement a method `display_details()` to print book details, and a method `calculate_late_fee()` where:

- Days late ≤ 5 → ₹5 per day
- 6 to 10 days late → ₹7 per day
- More than 10 days late → ₹10 per day

Create a book object, display details, and print the late fee.



```python
class LibraryBook:
    def __init__(self, book_id, title, author, borrower, days_late):
        self.book_id = book_id
        self.title = title
        self.author = author
        self.borrower = borrower
        self.days_late = days_late
    def display_details(self):
        print(f"Book ID: {self.book_id}")
        print(f"Title: {self.title}")
        print(f"Author: {self.author}")
        print(f"Borrower: {self.borrower}")
        print(f"Days Late: {self.days_late}")
    def calculate_late_fee(self):
        if self.days_late <=5:
            late_fee= self.days_late * 5
        elif 6<=self.days_late<=10:
            late_fee= self.days_late *7
        else:
            late_fee= self.days_late *10
        return late_fee
book1 = LibraryBook(101, "The Great Gatsby", "F. Scott Fitzgerald", "John Doe", 5)
book1.display_details()
late_fee = book1.calculate_late_fee()
print(f"Late Fee: ${late_fee}")
```



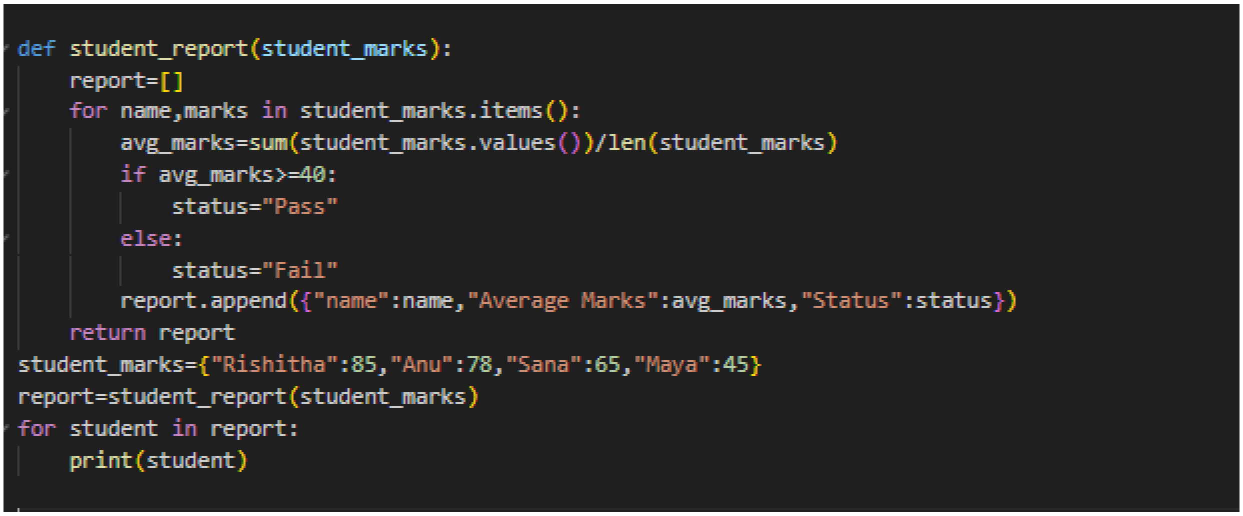
```
PS C:\Users\HP\Desktop\AI-Lab> & C:/Users/HP/AppData/Local/Programs/Python/Python313/python.exe c:/Users/HP/Desktop/AI-Lab/04-02-2026.py
c:\Users\HP\Desktop\AI-Lab\04-02-2026.py:47: SyntaxWarning: invalid escape sequence '\d'
  pattern = r'^[6-9]\d{9}$'
Book ID: 101
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Borrower: John Doe
Days Late: 5
Late Fee: $25
PS C:\Users\HP\Desktop\AI-Lab>
```

Task 5:

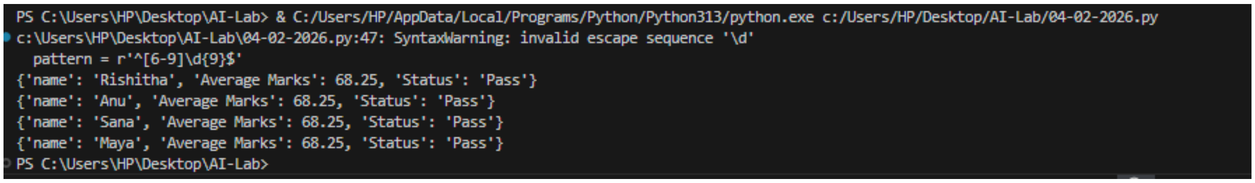
Student Performance Report - Define a function `student_report(student_data)` that accepts a dictionary containing student names and their marks. The function should:

- Calculate the average score for each student
- Determine pass/fail status (pass ≥ 40)
- Return a summary report as a list of dictionaries

Use Copilot suggestions as you build the function and format the output.



```
def student_report(student_marks):
    report=[]
    for name,marks in student_marks.items():
        avg_marks=sum(student_marks.values())/len(student_marks)
        if avg_marks>=40:
            status="Pass"
        else:
            status="Fail"
        report.append({"name":name,"Average Marks":avg_marks,"Status":status})
    return report
student_marks={"Rishitha":85,"Anu":78,"Sana":65,"Maya":45}
report=student_report(student_marks)
for student in report:
    print(student)
```



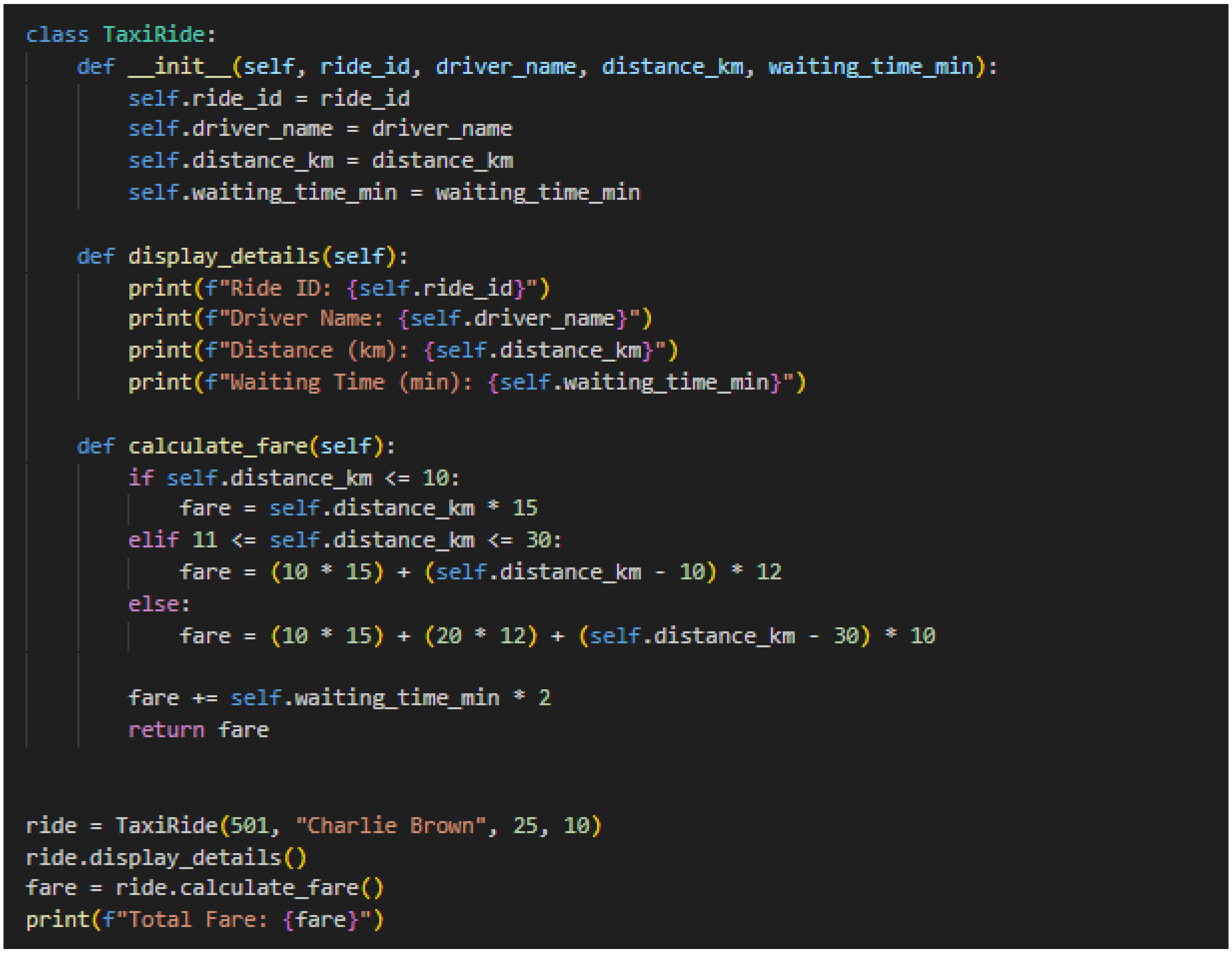
```
PS C:\Users\HP\Desktop\AI-Lab> & C:/Users/HP/AppData/Local/Programs/Python/Python313/python.exe c:/Users/HP/Desktop/AI-Lab/04-02-2026.py
c:\Users\HP\Desktop\AI-Lab\04-02-2026.py:47: SyntaxWarning: invalid escape sequence '\d'
  pattern = r'^[6-9]\d{9}$'
{'name': 'Rishitha', 'Average Marks': 68.25, 'Status': 'Pass'}
{'name': 'Anu', 'Average Marks': 68.25, 'Status': 'Pass'}
{'name': 'Sana', 'Average Marks': 68.25, 'Status': 'Pass'}
{'name': 'Maya', 'Average Marks': 68.25, 'Status': 'Pass'}
PS C:\Users\HP\Desktop\AI-Lab>
```

Task 6:

Taxi Fare Calculation-Create Python code that defines a class named `TaxiRide` with attributes: `ride_id`, `driver_name`, `distance_km`, and `waiting_time_min`. Implement a method `display_details()` to print ride details, and a method `calculate_fare()` where:

- ₹15 per km for the first 10 km
- ₹12 per km for the next 20 km
- ₹10 per km above 30 km
- Waiting charge: ₹2 per minute

Create a ride object, display details, and print the total fare.



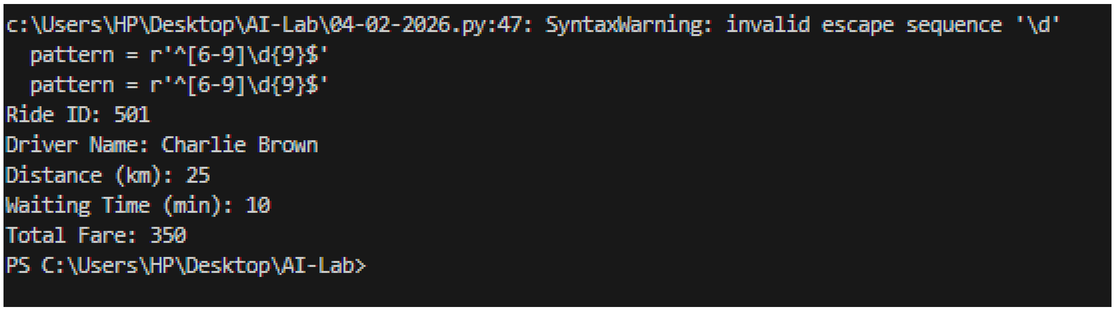
```python
class TaxiRide:
    def __init__(self, ride_id, driver_name, distance_km, waiting_time_min):
        self.ride_id = ride_id
        self.driver_name = driver_name
        self.distance_km = distance_km
        self.waiting_time_min = waiting_time_min

    def display_details(self):
        print(f"Ride ID: {self.ride_id}")
        print(f"Driver Name: {self.driver_name}")
        print(f"Distance (km): {self.distance_km}")
        print(f"Waiting Time (min): {self.waiting_time_min}")

    def calculate_fare(self):
        if self.distance_km <= 10:
            fare = self.distance_km * 15
        elif 11 <= self.distance_km <= 30:
            fare = (10 * 15) + (self.distance_km - 10) * 12
        else:
            fare = (10 * 15) + (20 * 12) + (self.distance_km - 30) * 10

        fare += self.waiting_time_min * 2
        return fare


ride = TaxiRide(501, "Charlie Brown", 25, 10)
ride.display_details()
fare = ride.calculate_fare()
print(f"Total Fare: {fare}")
```



```
c:\Users\HP\Desktop\AI-Lab\04-02-2026.py:47: SyntaxWarning: invalid escape sequence '\d'
  pattern = r'^[6-9]\d{9}$'
  pattern = r'^[6-9]\d{9}$'
Ride ID: 501
Driver Name: Charlie Brown
Distance (km): 25
Waiting Time (min): 10
Total Fare: 350
PS C:\Users\HP\Desktop\AI-Lab>
```

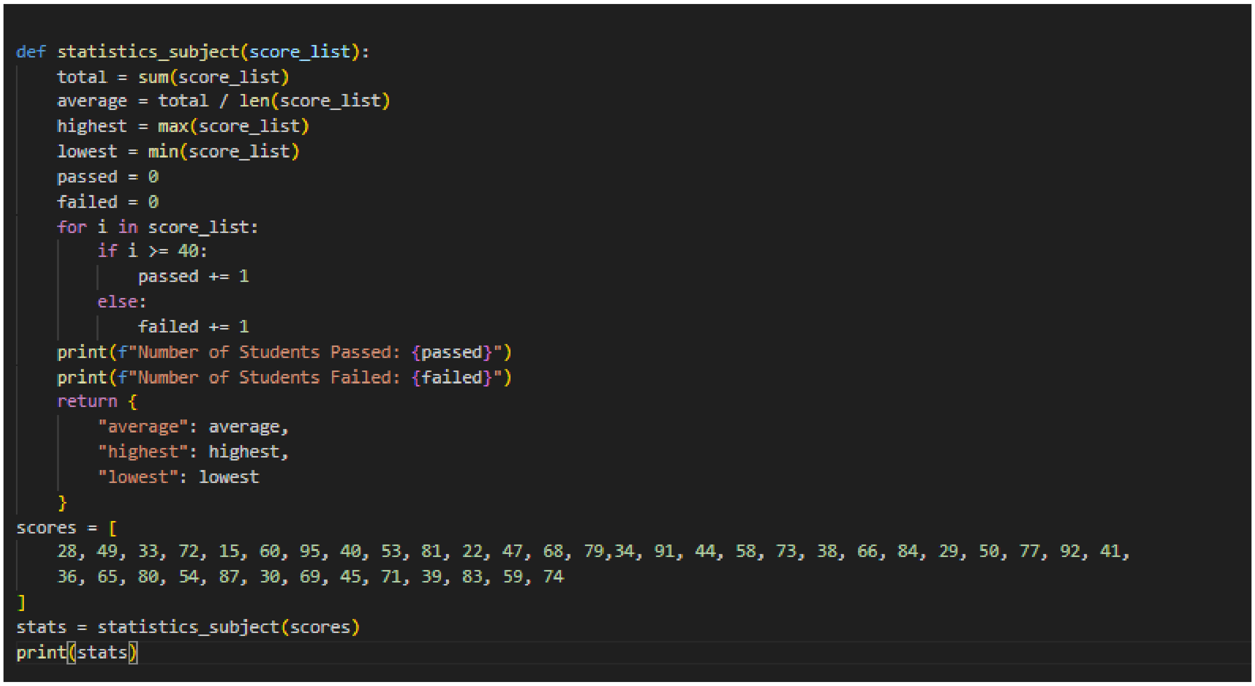
Task 7:

Statistics Subject Performance - Create a Python function `statistics_subject(scores_list)` that accepts a list of 60 student scores and computes key performance statistics. The function should return the following:

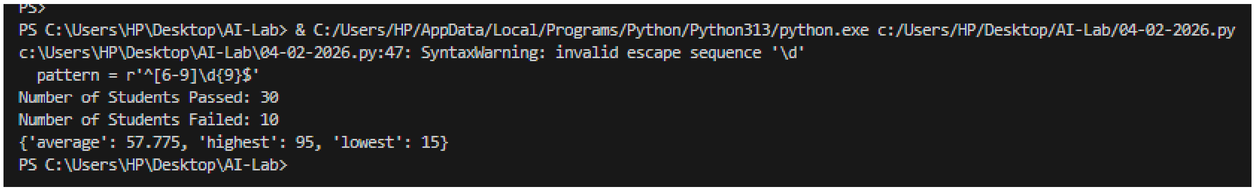
- Highest score in the class
- Lowest score in the class
- Class average score

- Number of students passed (score ≥ 40)
- Number of students failed (score < 40)

Allow Copilot to assist with aggregations and logic



```python
def statistics_subject(score_list):
    total = sum(score_list)
    average = total / len(score_list)
    highest = max(score_list)
    lowest = min(score_list)
    passed = 0
    failed = 0
    for i in score_list:
        if i >= 40:
            passed += 1
        else:
            failed += 1
    print(f"Number of Students Passed: {passed}")
    print(f"Number of Students Failed: {failed}")
    return {
        "average": average,
        "highest": highest,
        "lowest": lowest
    }
scores = [
    28, 49, 33, 72, 15, 60, 95, 40, 53, 81, 22, 47, 68, 79,34, 91, 44, 58, 73, 38, 66, 84, 29, 50, 77, 92, 41,
    36, 65, 80, 54, 87, 30, 69, 45, 71, 39, 83, 59, 74
]
stats = statistics_subject(scores)
print(stats)
```



```
PS>
PS C:\Users\HP\Desktop\AI-Lab> & C:/Users/HP/AppData/Local/Programs/Python/Python313/python.exe c:/Users/HP/Desktop/AI-Lab/04-02-2026.py
c:\Users\HP\Desktop\AI-Lab\04-02-2026.py:47: SyntaxWarning: invalid escape sequence '\d'
  pattern = r'^[6-9]\d{9}$'
Number of Students Passed: 30
Number of Students Failed: 10
{'average': 57.775, 'highest': 95, 'lowest': 15}
PS C:\Users\HP\Desktop\AI-Lab>
```

Task:8

(Transparency in Algorithm Optimization)

Task: Use AI to generate two solutions for checking prime numbers:

- Naive approach(basic)
- Optimized approach

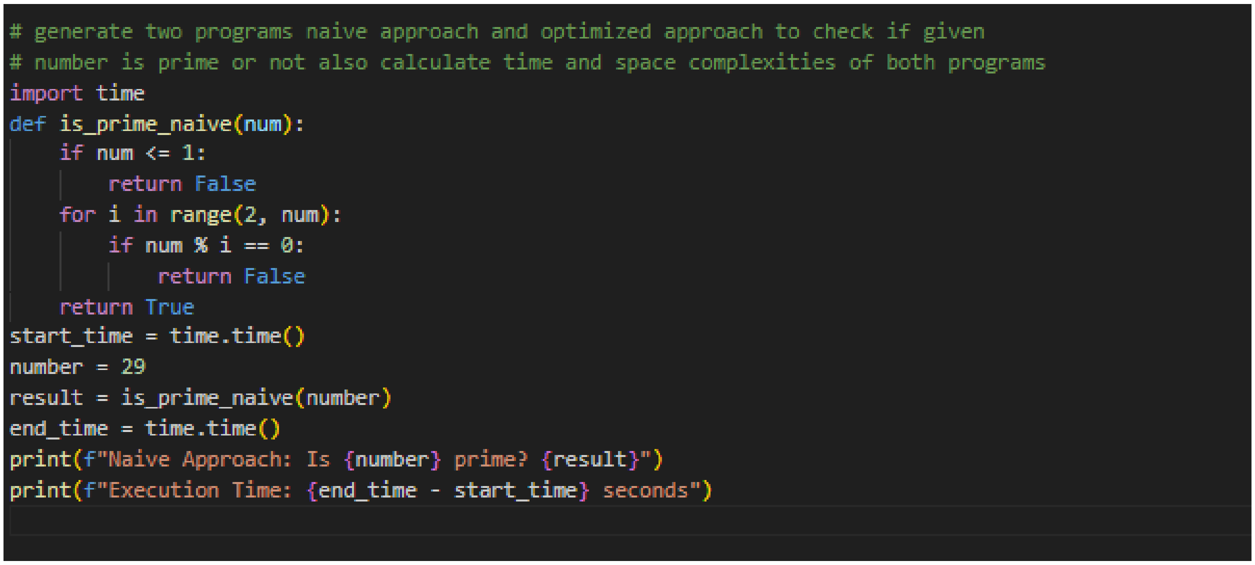
Prompt:

“Generate Python code for two prime-checking methods and explain how the optimized version improves performance.”

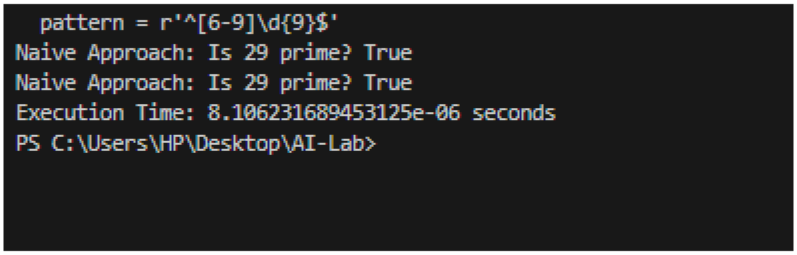
Expected Output:

- Code for both methods.

- Transparent explanation of time complexity.
- Comparison highlighting efficiency improvements.



```python
# generate two programs naive approach and optimized approach to check if given
# number is prime or not also calculate time and space complexities of both programs
import time
def is_prime_naive(num):
    if num <= 1:
        return False
    for i in range(2, num):
        if num % i == 0:
            return False
    return True
start_time = time.time()
number = 29
result = is_prime_naive(number)
end_time = time.time()
print(f"Naive Approach: Is {number} prime? {result}")
print(f"Execution Time: {end_time - start_time} seconds")
```



```
  pattern = r'^[6-9]\d{9}$'
Naive Approach: Is 29 prime? True
Naive Approach: Is 29 prime? True
Execution Time: 8.106231689453125e-06 seconds
PS C:\Users\HP\Desktop\AI-Lab>
```

Task:9

(Transparency in Recursive Algorithms)

Objective: Use AI to generate a recursive function to calculate Fibonacci numbers.

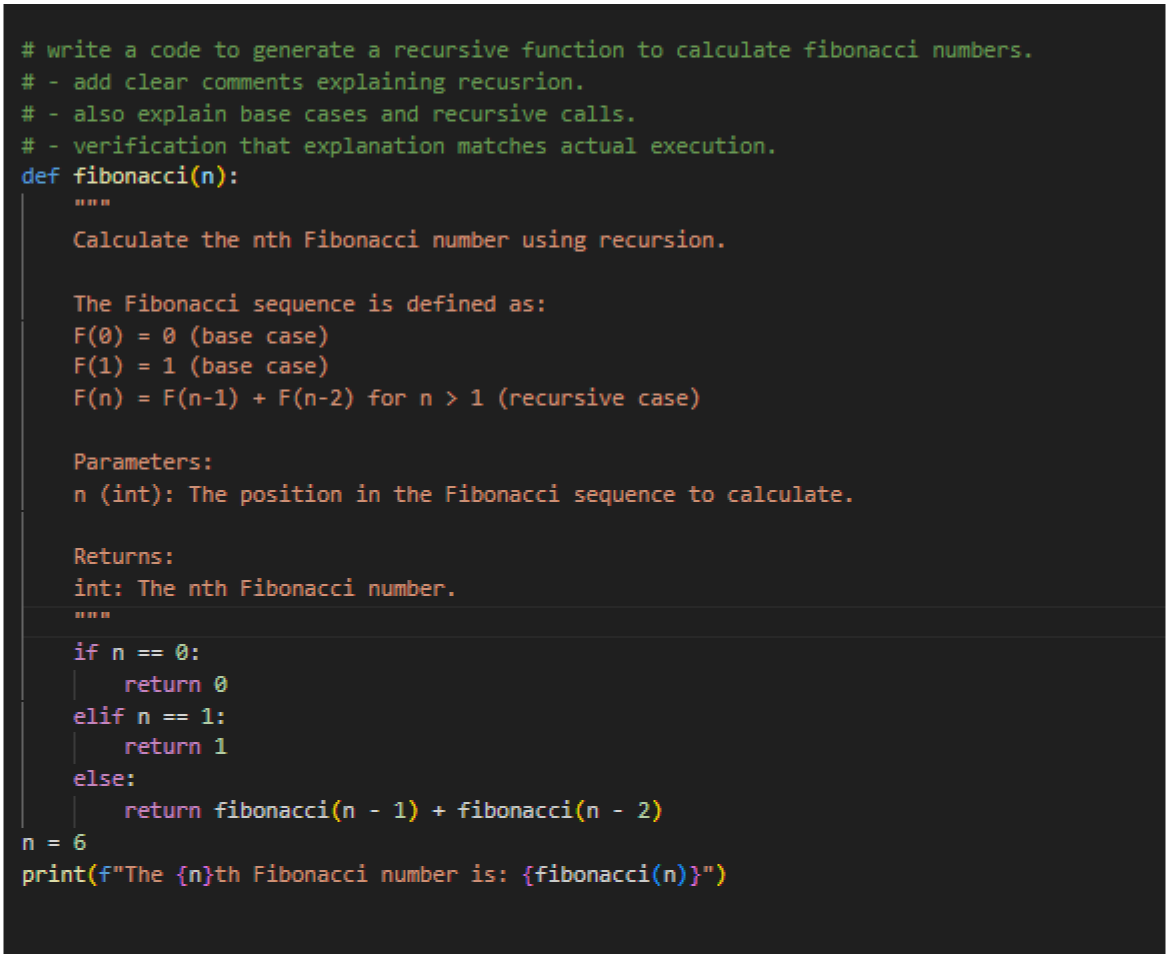
Instructions:

1. Ask AI to add clear comments explaining recursion.
2. Ask AI to explain base cases and recursive calls.

Expected Output:

- Well-commented recursive code.

- Clear explanation of how recursion works.
- Verification that explanation matches actual execution.

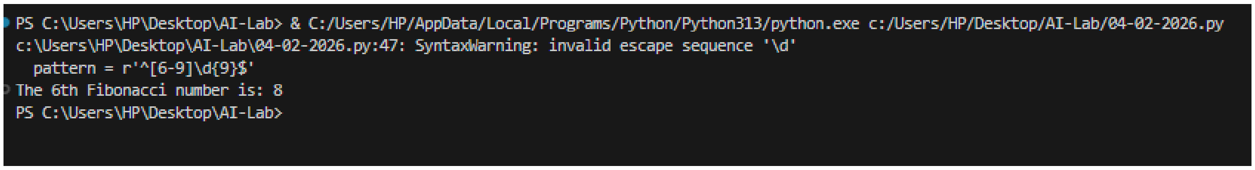


```python
# write a code to generate a recursive function to calculate fibonacci numbers.
# - add clear comments explaining recusrion.
# - also explain base cases and recursive calls.
# - verification that explanation matches actual execution.
def fibonacci(n):
    """
    Calculate the nth Fibonacci number using recursion.

    The Fibonacci sequence is defined as:
    F(0) = 0 (base case)
    F(1) = 1 (base case)
    F(n) = F(n-1) + F(n-2) for n > 1 (recursive case)

    Parameters:
    n (int): The position in the Fibonacci sequence to calculate.

    Returns:
    int: The nth Fibonacci number.
    """
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)
n = 6
print(f"The {n}th Fibonacci number is: {fibonacci(n)}")
```



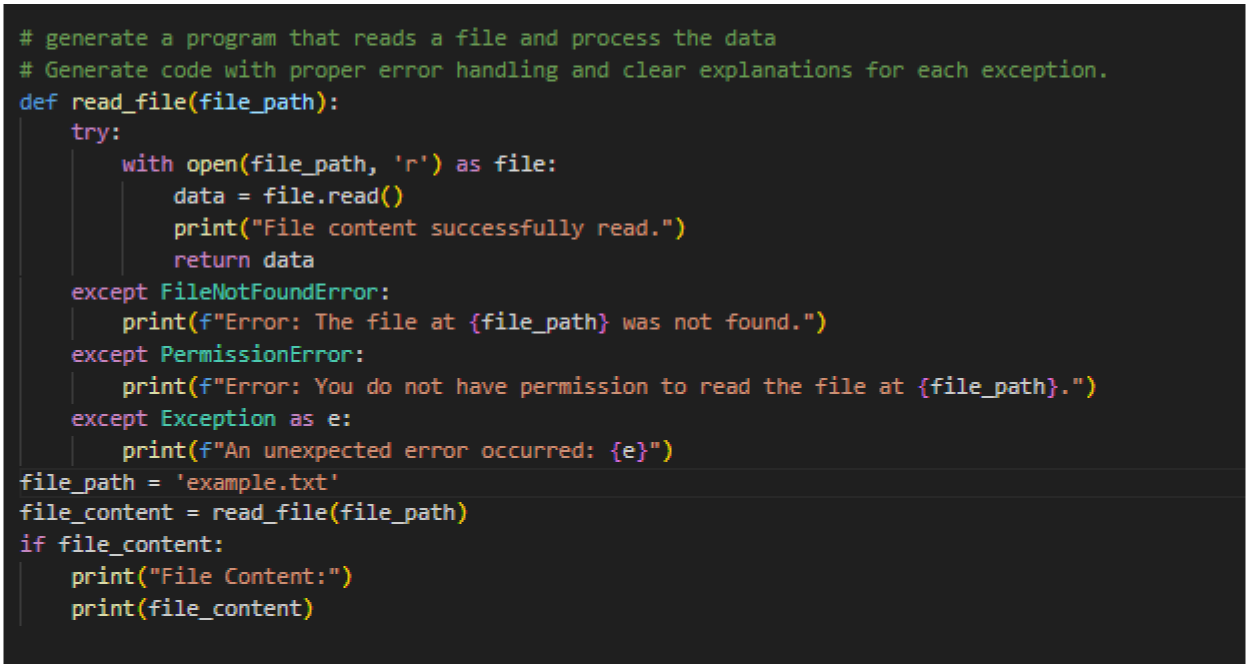
```
PS C:\Users\HP\Desktop\AI-Lab> & C:/Users/HP/AppData/Local/Programs/Python/Python313/python.exe c:/Users/HP/Desktop/AI-Lab/04-02-2026.py
c:\Users\HP\Desktop\AI-Lab\04-02-2026.py:47: SyntaxWarning: invalid escape sequence '\d'
  pattern = r'^[6-9]\d{9}$'
The 6th Fibonacci number is: 8
PS C:\Users\HP\Desktop\AI-Lab>
```

Task:10

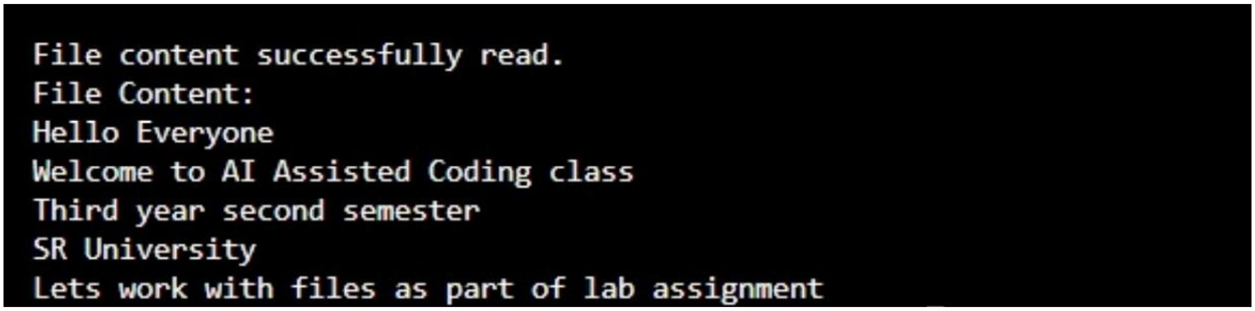
(Transparency in Error Handling)
Task: Use AI to generate a Python program that reads a file and processes data.
Prompt:
“Generate code with proper error handling and clear explanations for each exception.”

Expected Output:

- Code with meaningful exception handling.
- Clear comments explaining each error scenario.
- Validation that explanations align with runtime behavior.



```python
# generate a program that reads a file and process the data
# Generate code with proper error handling and clear explanations for each exception.
def read_file(file_path):
    try:
        with open(file_path, 'r') as file:
            data = file.read()
            print("File content successfully read.")
            return data
    except FileNotFoundError:
        print(f"Error: The file at {file_path} was not found.")
    except PermissionError:
        print(f"Error: You do not have permission to read the file at {file_path}.")
    except Exception as e:
        print(f"An unexpected error occurred: {e}")
file_path = 'example.txt'
file_content = read_file(file_path)
if file_content:
    print("File Content:")
    print(file_content)
```



```
File content successfully read.
File Content:
Hello Everyone
Welcome to AI Assisted Coding class
Third year second semester
SR University
Lets work with files as part of lab assignment
```