

LAB-8.3

-2303A51197

Task 1: Email Validation using TDD

Scenario: You are developing a user registration system that requires reliable email input validation.

Prompt used:

#create a python function for validating email in a registration system

#email must contain one @ symbol and at least one .

#email should not begin or end with special characters

#multiple @ symbols should not be allowed.

#return true if the is valid emil, otherwise return false.

//OUTOUT:

```
C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig\lab_8.3.py will in a registration system
1 #email must contain one @ symbol and at least one .
2 #email should not begin or end with special characters
3 #multiple @ symbols should not be allowed.
4 #return true if the is valid emil, otherwise return false.
5 #don't give comments in the code.
6
7 def validate_email(email):
8     if email.count('@') != 1:
9         return False
10    if email.startswith('@') or email.endswith('@'):
11        return False
12    if email.startswith('..') or email.endswith('..'):
13        return False
14    if '.' not in email.split('@')[1]:
15        return False
16    return True
17 #display output
18 email = input("Enter your email: ")
19 if validate_email(email):
20     print("Valid email")
21 else:
22     print("Invalid email")

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRAKASH\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/U
Enter your email: prakashkommarajul@gmail.com
Valid email
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRAKASH\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/U
Enter your email: njcsbhjdb
Invalid email
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig>
```

Task 2: Grade Assignment using Loops

Scenario: You are building an automated grading system for an online examination platform.

Prompt used:

#create a python function for assigning grades automatically

```
#print A for 90-100, B for 80-89, C for 70-79, D for 60-69, F for below 60
```

```
#make sure boundary values like 60, 70, 80, 90 are included
```

```
//OUTOUT:
```

The screenshot shows a VS Code interface with a dark theme. On the left is a file tree with files like `lab_83.py`, `lab_83.java`, and `lab_83.java`. The main editor window contains Python code for assigning grades based on scores. The terminal at the bottom shows the execution of the script and its output for different input scores.

```
lab_83.py
24 #create a python function for assigning grades automatically
25 #print A for 90-100, B for 80-89, C for 70-79, D for 60-69, F for below 60
26 #make sure boundary values like 60, 70, 80, 90 are included
27 def assign_grade(score):
28     if score >= 90 and score <= 100:
29         return 'A'
30     elif score >= 80 and score < 90:
31         return 'B'
32     elif score >= 70 and score < 80:
33         return 'C'
34     elif score >= 60 and score < 70:
35         return 'D'
36     elif score < 60:
37         return 'F'
38     else:
39         return 'Invalid score'
40 #display output
41 score = float(input("Enter your score: "))
42 grade = assign_grade(score)
43 print(f"Your grade is: {grade}")

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\PRakash\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRakash\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/PRakash/OneDrive/Desktop\lab_83.py"
● PS C:\Users\PRakash\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRakash\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/PRakash/OneDrive/Desktop\lab_83.py"
Enter your score: 35
Your grade is: F
● PS C:\Users\PRakash\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRakash\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/PRakash/OneDrive/Desktop\lab_83.py"
Enter your score: 88
Your grade is: B
● PS C:\Users\PRakash\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRakash\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/PRakash/OneDrive/Desktop\lab_83.py"
Enter your score: 100
Your grade is: A
○ PS C:\Users\PRakash\OneDrive\Desktop\Ai assisted codig>
```

Task 3: Sentence Palindrome Checker

Scenario: You are developing a text-processing utility to analyze sentences.

Prompt used:

```
#create a python program to verify whether a sentence is palindrome
```

```
#ignore spaces, punctuation marks and letter case.
```

The screenshot shows a VS Code interface with a dark theme. On the left is a file tree with files like `lab_83.py`, `app.log`, and `LAB_3.4.java`. The main editor window contains Python code for checking if a sentence is a palindrome. The terminal at the bottom shows the execution of the script and its output for different input sentences.

```
lab_83.py
42 # grade = assign_grade(score)
43 # print(f"Your grade is: {grade}")
44
45 #create a python program to verify whether a sentence is palindrome
46 #ignore spaces, punctuation marks and letter case.
47 import string
48 def is_palindrome(sentence):
49     cleaned_sentence = ''.join(char.lower() for char in sentence if char.isalnum())
50     return cleaned_sentence == cleaned_sentence[::-1]
51 #display output
52 sentence = input("Enter a sentence: ")
53 if is_palindrome(sentence):
54     print("The sentence is a palindrome.")
55 else:
56     print("The sentence is not a palindrome.")

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\PRakash\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRakash\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/PRakash/OneDrive/Desktop\lab_83.py"
PS C:\Users\PRakash\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRakash\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/PRakash/OneDrive/Desktop\lab_83.py"
○ Enter a sentence: madam
The sentence is a palindrome.
PS C:\Users\PRakash\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRakash\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/PRakash/OneDrive/Desktop\lab_83.py"
Enter a sentence: i am back
The sentence is not a palindrome.
PS C:\Users\PRakash\OneDrive\Desktop\Ai assisted codig>
```

Task 4: ShoppingCart Class

Scenario: You are designing a basic shopping cart module for an e-commerce application.

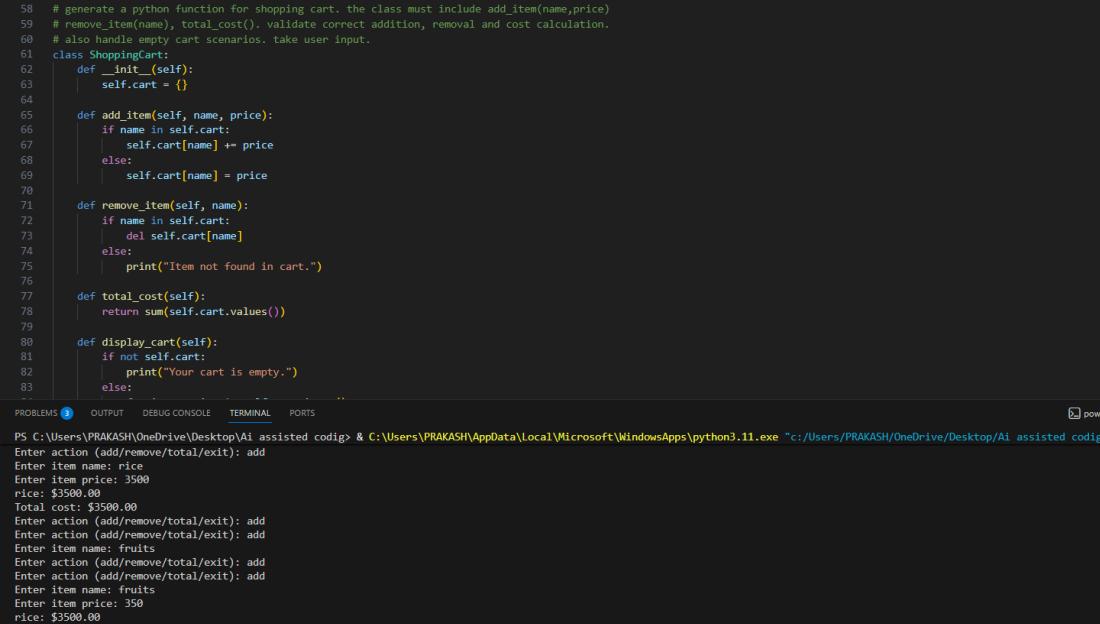
Prompt used:

```
# generate a python function for shopping cart. the class must include add_item(name,price)
```

remove_item(name), total_cost(). validate correct addition, removal and cost calculation.

also handle empty cart scenarios. take user input

//OUTOUT:



The screenshot shows the Visual Studio Code interface with the following details:

- OPEN EDITORS:** Shows multiple files: lab_8.3.py, lab_8.3.py, AI ASSISTED CODIG, app.log, cp_lab.java, lab_3.3.py, LAB_3.java, lab_5.4.py, lab_6.3.py, lab_6.3, lab_6.4.py, lab_7.3.py, lab_8.3.py (highlighted in blue), LAB_EXAM.py, and lab1.py.
- PROBLEMS:** Shows 3 errors.
- OUTPUT:** Displays command-line output from a PowerShell terminal:

```
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRAKASH\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/PRAKASH/OneDrive/Desktop/Ai assisted codig/lab_8.3.py"
Enter action (add/remove/total/exit): add
Enter item name: rice
Enter item price: 3500
rice: $3500.00
Total cost: $3500.00
Enter action (add/remove/total/exit): add
Enter action (add/remove/total/exit): add
Enter item name: fruits
Enter item price: 350
fruits: $350.00
Total cost: $3850.00
Enter action (add/remove/total/exit): total
Total cost: $3850.00
Enter action (add/remove/total/exit): exit
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig>
```
- TERMINAL:** Shows the PowerShell prompt and command history.
- OUTLINE, TIMELINE, PROJECTS, RUN CONFIGURATION:** Standard VS Code navigation and management tools.

Task 5: Date Format Conversion

Scenario: You are creating a utility function to convert date formats for reports.

Prompt used:

```
#create a python function to change date format
```

#convert date from "YYYY-MM-DD" format into "DD-MM-YYYY" format

//OUTOUT:

The screenshot shows a code editor window with a dark theme. At the top, there's a status bar with file paths: 'C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig\lab_6.4.py'. Below the status bar is the code editor area containing a Python script. The script defines a function `change_date_format` that takes a date string in "YYYY-MM-DD" format and returns it in "DD-MM-YYYY" format. It includes error handling for invalid formats. The code is color-coded with syntax highlighting. At the bottom of the editor, there are tabs for 'PROBLEMS' (with 3 errors), 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is selected. In the terminal pane, there are three command-line entries:

- PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> C:\Users\PRAKASH\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/U...
Enter a date in YYYY-MM-DD format: 2005-12-06
06-12-2005
- PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRAKASH\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/...
Enter a date in YYYY-MM-DD format: 1898-06-12
12-06-1898
- PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig>

Explanation:

1. The first snippet is a function to validate email addresses based on specific criteria.
2. The second snippet is a function to calculate letter grades based on numerical scores.
3. The third snippet checks if a given sentence is a palindrome, ignoring spaces and punctuation.
4. The fourth snippet defines a ShoppingCart class that allows users to add and remove items, calculate total cost, and display the cart contents.
5. The fifth snippet is a function that converts a date from "YYYY-MM-DD" format to "DD-MM-YYYY" format.