

## AI-ASSISTED CODING

Assignment Number:8.3

Name – sai akshith

BATCH – 45

Task 1: Email Validation using TDD

PROMT:

```
# valid email address You are developing a user registration system that requires reliable  
email input validation.it must cointain "@" and "." and should not contain any spaces and  
not allow multiple "@"and should cointain only small letters and should cointain gmail.com  
or yahoo.com or outlook.com
```

CODE:

```
email = input("Enter your email address: ")  
  
if "@" in email and "." in email and " " not in email and email.count("@") == 1 and  
email.islower() and (email.endswith("@gmail.com") or email.endswith("@yahoo.com") or  
email.endswith("@outlook.com")):  
  
    print("Valid email address.")  
  
else:  
  
    print("Invalid email address.")
```

OUTPUT:

```

File Edit Selection View Go Run Terminal Help < > Q AI-PROGRAMMING
EXPLORER AI-PROGRAMMING
exp 1.py
exp 2.py
exp 4-5.py
exp 5.py
exp 6-2.py
exp 6-3.py
exp 7-5.py
exp 8-3.py
exp4-3.py
user_activity.log
... exp 8-3.py ...
1 # valid email address You are developing a user registration system that requires reliable email input validation.it must contain
2 email = input("Enter your email address: ")
3 if "@" in email and "." in email and " " not in email and email.count("@") == 1 and email.islower() and (email.endswith("@gmail.com")
4 | print("Valid email address.")
5 else:
6 | print("Invalid email address.")
7
8 # generate automated grading system for an online examination platform
9 # where the grading system is like
10 # 90-100 → A
11 # 80-89 → B
12 # 70-79 → C
13 # 60-69 → D
14 # Below 60 → F
15 # Include boundary values (60, 70, 80, 90)
16 # Include invalid inputs such as -5, 105, "eighty"
17 score = input("Enter the exam score: ")
18 try:
19 | score = int(score)
20 | if 90 <= score <= 100:
21 | | print("Grade: A")
PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING & C:/Python314/python.exe "C:/Users/SAIAKSHITH/OneDrive/Documents/AI-PROGRAMMING/exp 8-3.py"
Enter your email address: saiaakshithballe@gmail.com
Valid email address.
Enter the exam score: 98
Grade: A
Enter a sentence: sas
The sentence is a palindrome.
Total cost: $199.98
Total cost after removal: $99.99
Item 'Smartphone' not found in the cart.
Converted date: 2024-12-25
PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING []

```

## Task 2: Grade Assignment using Loops

PROMT:

```

# generate automated grading system for an online examination platform
# where the grading system is like
# 90–100 → A
# 80–89 → B
# 70–79 → C
# 60–69 → D
# Below 60 → F
# Include boundary values (60, 70, 80, 90)
# Include invalid inputs such as -5, 105, "eighty"

```

CODE:

```
score = input("Enter the exam score: ")
```

```
try:
```

```

score = int(score)

if 90 <= score <= 100:
    print("Grade: A")

elif 80 <= score < 90:
    print("Grade: B")

elif 70 <= score < 80:
    print("Grade: C")

elif 60 <= score < 70:
    print("Grade: D")

elif score < 60:
    print("Grade: F")

else:
    print("Invalid score. Score must be between 0 and 100.")

except ValueError:
    print("Invalid input. Please enter a numeric score.")

```

## OUTPUT:

```

File Edit Selection View Go Run Terminal Help ⏪ ⏩ Q AI-PROGRAMMING
EXPLORER AI-PROGRAMMING
exp 1.py
exp 2.py
exp 4-5.py
exp 5.py
exp 6-2.py
exp 7-5.py
exp 8-3.py
exp4-3.py
user_activity.log
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING> & c:/Python314/python.exe "c:/Users/SAIAKSHITH/OneDrive/Documents/AI-PROGRAMMING/exp 8-3.py"
Enter your email address: salakshithballe@gmail.com
Valid email address.
Enter the exam score: 95
Grade: A

```

### Task 3: Sentence Palindrome Checker

PROMT:

```
# generate a sentence palindrome checker that checks if a given sentence is a palindrome or not. A palindrome is a word, phrase, number, or other sequence of characters that reads the same forward and backward (ignoring spaces, punctuation, and capitalization).
```

CODE:

```
import string

def is_palindrome(sentence):

    cleaned_sentence = ''.join(char for char in sentence if char.isalnum()).lower()

    return cleaned_sentence == cleaned_sentence[::-1]

sentence = input("Enter a sentence: ")

if is_palindrome(sentence):

    print("The sentence is a palindrome.")

else:

    print("The sentence is not a palindrome.")
```

OUTPUT:

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows files in the "AI-PROGRAMMING" folder: exp 1.py, exp 2.py, exp 4-5.py, exp 5.py, exp 6-2.py, exp 7-5.py, exp 8-3.py (selected), exp4-3.py, and user\_activity.log.
- Code Editor:** Displays a Python script named "exp 8-3.py". The code includes a function to check if a sentence is a palindrome and a basic shopping cart module.
- Terminal:** Shows the command PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING & C:\Python314\python.exe "c:/Users/SAIAKSHITH/OneDrive/Documents/AI-PROGRAMMING/exp 8-3.py". It prompts for an email address, asks for an exam score (98), and displays output related to a shopping cart.
- Status Bar:** Shows file path (PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING), line count (Ln 47), column count (Col 44), spaces (Spaces: 4), encoding (UTF-8), CR/LF (CR LF), Python 3.14.2, and system status (ENG IN, 11-02-2026).

## Task 4: ShoppingCart Class

PROMT:

```
# generate a basic shopping cart module for an e-commerce application.

# Class must include the following methods:

# add_item(name, price)

# remove_item(name)

# total_cost()

# Validate correct addition, removal, and cost calculation

# Handle empty cart scenarios
```

CODE:

```
class ShoppingCart:
```

```
    def __init__(self):
        self.cart = {}
```

```
def add_item(self, name, price):
    if name in self.cart:
        self.cart[name] += price
    else:
        self.cart[name] = price

def remove_item(self, name):
    if name in self.cart:
        del self.cart[name]
    else:
        print(f"Item '{name}' not found in the cart.")

def total_cost(self):
    return sum(self.cart.values())

# Example usage
cart = ShoppingCart()
cart.add_item("Laptop", 999.99)
cart.add_item("Headphones", 199.99)
print(f"Total cost: ${cart.total_cost():.2f}")
cart.remove_item("Headphones")
print(f"Total cost after removal: ${cart.total_cost():.2f}")
cart.remove_item("Smartphone")
```

OUTPUT:

```
class ShoppingCart:
    def __init__(self):
        self.cart = {}

    def add_item(self, name, price):
        if name in self.cart:
            self.cart[name] += price
        else:
            self.cart[name] = price

    def remove_item(self, name):
        if name in self.cart:
            del self.cart[name]
        else:
            print(f"Item '{name}' not found in the cart.")

    def total_cost(self):
        return sum(self.cart.values())

# Example usage
cart = ShoppingCart()
cart.add_item("apple", 100)
cart.add_item("banana", 150)
cart.remove_item("apple")
print(cart.total_cost())
# Output: 150
```

## Task 5: Date Format Conversion

PROMT:

```
# create a utility function to convert date formats for reports. The function should take a date string in the format "DD/MM/YYYY" and convert it to "YYYY-MM-DD".
```

CODE:

```
def convert_date_format(date_str):
    try:
        day, month, year = date_str.split('/')
        return f"{year}-{month.zfill(2)}-{day.zfill(2)}"
    except ValueError:
        return "Invalid date format. Please use 'DD/MM/YYYY'."

# Example usage
date_input = "25/12/2024"
converted_date = convert_date_format(date_input)
print(f"Converted date: {converted_date}")
```

## OUTPUT:

```
File Edit Selection View Go Run Terminal Help <- > Q AI-PROGRAMMING
EXPLORER AI-PROGRAMMING
exp 1.py
exp 2.py
exp 4-5.py
exp 5.py
exp 6-2.py
exp 7-5.py
exp 8-3.py
exp4-3.py
user_activity.log

exp 8-3.py
76 # Create a utility function to convert date formats for reports. The function should take a date string in the format "DD/MM/YYYY"
77 def convert_date_format(date_str):
78     try:
79         day, month, year = date_str.split('/')
80         return f"({year}-{month.zfill(2)})-{day.zfill(2)}"
81     except ValueError:
82         return "Invalid date format. Please use 'DD/MM/YYYY'."
83 # Example usage
84 date_input = "25/12/2024"
85 converted_date = convert_date_format(date_input)
86 print(f"Converted date: {converted_date}")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
+ v ... | x Python Python
PS C:\Users\SAIASHITH\OneDrive\Documents\AI-PROGRAMMING> & c:/Python314/python.exe "c:/Users/SAIASHITH/OneDrive/Documents/AI-PROGRAMMING/exp 8-3.py"
Enter your email address: saiaashithballe@gmail.com
Valid email address.
Enter the exam score: 98
Grade: A
Enter a sentence: sas
The sentence is a palindrome.
Total cost: $1199.99
Total cost after removal: $999.99
Item 'Smartphone' not found in the cart.
Converted date: 2024-12-25
PS C:\Users\SAIASHITH\OneDrive\Documents\AI-PROGRAMMING>

Ln 81, Col 1 (76 selected) Spaces: 4 UTF-8 CRLF {} Python 3.14.2 Go Live ENG IN 10:19 11-02-2026
```