

AI Assisted Coding Assignment- 8.2

RUDROJU RUPA SRI

2303A51918

BATCH-30

Task 1 – Test-Driven Development for Even/Odd Number Validator

- Use AI tools to first generate test cases for a function `is_even(n)` and then implement the function so that it satisfies all generated tests.

Requirements:

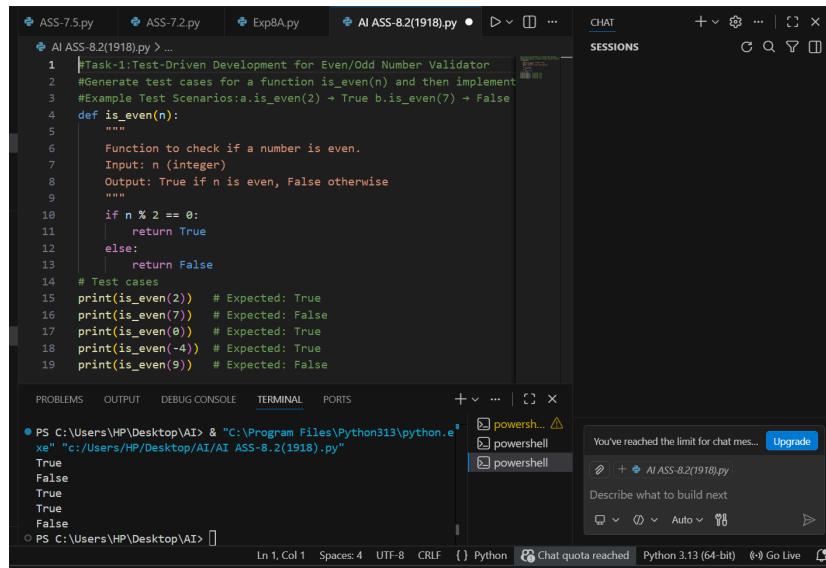
- Input must be an integer
- Handle zero, negative numbers, and large integers

Example Test Scenarios:

`is_even(2) → True`
`is_even(7) → False`
`is_even(0) → True`
`is_even(-4) → True`
`is_even(9) → False`

Expected Output -1

- A correctly implemented `is_even()` function that passes all AI-generated test cases



The screenshot shows a code editor with several tabs open. The active tab is `AI ASS-8.2(1918).py`, which contains the following Python code:

```
1  #Task-1:Test-Driven Development for Even/Odd Number Validator
2  #Generate test cases for a function is_even(n) and then implement
3  #Example Test Scenarios:a.is_even(2) → True b.is_even(7) → False
4  def is_even(n):
5      """
6          Function to check if a number is even.
7          Input: n (integer)
8          Output: True if n is even, False otherwise
9      """
10     if n % 2 == 0:
11         return True
12     else:
13         return False
14
15 # Test cases
16 print(is_even(2)) # Expected: True
17 print(is_even(7)) # Expected: False
18 print(is_even(0)) # Expected: True
19 print(is_even(-4)) # Expected: True
20 print(is_even(9)) # Expected: False
```

Below the code editor is a terminal window showing the output of running the script:

```
PS C:\Users\HP\Desktop\AI> & "C:\Program Files\Python313\python.exe" "c:/Users/HP/Desktop/AI/AI ASS-8.2(1918).py"
True
False
True
True
False
```

The terminal also shows the path `C:\Users\HP\Desktop\AI>`. At the bottom of the terminal, there is a message: "You've reached the limit for chat mes... [Upgrade]".

Task 2 – Test-Driven Development for String Case Converter

- Ask AI to generate test cases for two functions:
- `to_uppercase(text)`
- `to_lowercase(text)`

Requirements:

- Handle empty strings
- Handle mixed-case input
- Handle invalid inputs such as numbers or None

Example Test Scenarios:

`to_uppercase("ai coding") → "AI CODING"`

`to_lowercase("TEST") → "test"`

`to_uppercase("") → ""`

`to_lowercase(None) → Error or safe handling`

Expected Output -2

- Two string conversion functions that pass all AI-generated test cases with safe input handling.

A screenshot of the Visual Studio Code interface. The left pane shows the code editor with Python files: `AI ASS-8.2(1918).py`, `AI ASS-7.5.py`, and `AI ASS-7.2.py`. The `AI ASS-8.2(1918).py` file contains two functions: `to_uppercase` and `to_lowercase`. The `to_uppercase` function handles None input by returning an error message. The `to_lowercase` function also handles None input by returning an error message. The right pane shows the terminal output, which includes a message about reaching the chat limit and some PowerShell session logs. The status bar at the bottom indicates the file is Python 3.13 (64-bit).

```
AI ASS-8.2(1918).py
...
# AI CODING
# to_uppercase("ai coding") → "AI CODING"
# to_lowercase("TEST") → "test"
# to_uppercase("") → ""
# to_lowercase(None) → Error or safe handling
def to_uppercase(text):
    """
    Function to convert a string to uppercase.
    Input: text (string)
    Output: Uppercase version of the input string
    """
    if text is None:
        return "Error: Input cannot be None"
    return text.upper()
def to_lowercase(text):
    """
    Function to convert a string to lowercase.
    Input: text (string)
    Output: Lowercase version of the input string
    """
    if text is None:
        return "Error: Input cannot be None"
    return text.lower()
# Test cases
print(to_uppercase("ai coding")) # Expected: "AI CODING"
print(to_lowercase("TEST")) # Expected: "test"
print(to_uppercase("")) # Expected: ""
print(to_lowercase(None)) # Expected: "Error: Input cannot be None"
```

Task 3 – Test-Driven Development for List Sum Calculator

- Use AI to generate test cases for a function `sum_list(numbers)` that calculates the sum of list elements.

Requirements:

- Handle empty lists
- Handle negative numbers
- Ignore or safely handle non-numeric values

Example Test Scenarios:

`sum_list([1, 2, 3]) → 6`

`sum_list([]) → 0`

`sum_list([-1, 5, -4]) → 0`

`sum_list([2, "a", 3]) → 5`

Expected Output 3

- A robust list-sum function validated using AI-generated test cases.

A screenshot of the Visual Studio Code interface. The left pane shows the code editor with Python files: `AI ASS-8.2(1918).py`, `AI ASS-7.5.py`, `AI ASS-7.2.py`, `Exp8.py`, and `AI ASS-4.5.py`. The `AI ASS-8.2(1918).py` file contains a `sum_list` function that handles empty lists, negative numbers, and non-numeric values by ignoring them. The right pane shows the terminal output, which includes a message about reaching the chat limit and some PowerShell session logs. The status bar at the bottom indicates the file is Python 3.13 (64-bit).

```
AI ASS-8.2(1918).py
...
# Generate test cases for a function sum_list(numbers) and then implement the function so that it satisfies all generated tests.
# Requirements:
# 1. Handle empty lists
# 2. Handle negative numbers
# 3. Handle non-numeric values
# 4. Ignore or safely handle non-numeric values
# Example Test Scenarios:
# 1. sum_list([1, 2, 3]) = 6
# 2. sum_list([]) = 0
# 3. sum_list([-1, 5, -4]) = 0
# 4. sum_list([2, "a", 3]) = 5
# Test cases
def sum_list(numbers):
    """
    Function to calculate the sum of numeric elements in a list.
    Input: numbers (list)
    Output: Sum of numeric values (ignores non-numeric values)
    """
    if not isinstance(numbers, list):
        raise ValueError("Error: Input must be a list")
    total = 0
    for item in numbers:
        if isinstance(item, (int, float)):
            total += item
            # Non-numeric values are ignored
    return total
# Test cases
print(sum_list([1, 2, 3])) # Expected: 6
print(sum_list([])) # Expected: 0
print(sum_list([-1, 5, -4])) # Expected: 0
print(sum_list([2, "a", 3])) # Expected: 5
print(sum_list(["x", None, 4])) # Expected: 4
print(sum_list([223])) # Expected: Error: Input must be a list
```

Task 4 – Test Cases for Student Result Class

- Generate test cases for a StudentResult class with the following methods:

- add_marks(mark)
- calculate_average()
- get_result()

Requirements:

- Marks must be between 0 and 100
- Average $\geq 40 \rightarrow$ Pass, otherwise Fail

Example Test Scenarios:

Marks: [60, 70, 80] \rightarrow Average: 70 \rightarrow Result: Pass

Marks: [30, 35, 40] \rightarrow Average: 35 \rightarrow Result: Fail

Marks: [-10] \rightarrow Error

Expected Output -4

- A fully functional StudentResult class that passes all AI-generated test

```
AI ASS-8.2(1918).py ...
87 # Task-4: Test Cases For StudentResult Class
88 # Generate test cases for a StudentResult class with methods:
89 # 1. add_marks()
90 # 2. calculate_average()
91 # 3. get_result()
92
93 # Requirements:
94 # 1. Marks must be between 0 and 100
95 # 2. Average >= 40 = Pass, otherwise Fail
96
97 # Example Test Scenarios:
98 # a. Marks: [60, 70, 80] + Average: 70 + Result: Pass
99 # b. Marks: [30, 35, 40] + Average: 35 + Result: Fail
100 # c. Marks: [-10] = Error
101
102
103 class StudentResult:
104     """
105         Class to store student marks and calculate result.
106     """
107     def __init__(self):
108         self.marks = []
109     def add_marks(self, mark):
110         """
111             Adds a mark to the list,
112             Mark must be between 0 and 100.
113         """
114         if not isinstance(mark, (int, float)):
115             return "Error: Marks must be numeric"
116         if mark < 0 or mark > 100:
117             return "Error: Marks must be between 0 and 100"
118         self.marks.append(mark)
119     def calculate_average(self):
120         """
121             Calculates average of marks.
122         """
123         if len(self.marks) == 0:
124             return 0
125
126
127     def get_result(self):
128         """
129             Returns Pass if average >= 40 else Fail.
130         """
131         avg = self.calculate_average()
132         if avg >= 40:
133             return "Pass"
134         else:
135             return "Fail"
136
137 # Test Case
138 # Test Case 1: Pass Scenario
139 student1 = StudentResult()
140 student1.add_marks(60)
141 print(student1.calculate_average()) # Expected: 70.0
142 print(student1.get_result()) # Expected: Pass
143
144 student2 = StudentResult()
145 student2.add_marks(30)
146 student2.add_marks(35)
147 student2.add_marks(40)
148 print(student2.calculate_average()) # Expected: 35.0
149 print(student2.get_result()) # Expected: Fail
150
151 student3 = StudentResult()
152 print(student3.add_marks(-10)) # Expected: Error
153 print(student3.add_marks(120)) # Expected: Error
154
155 student4 = StudentResult()
156 print(student4.calculate_average()) # Expected: 0
157 print(student4.get_result()) # Expected: Fail
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\HP\Desktop\AI> & "C:\Program Files\Python311\python.exe" "C:/Users/HP/Desktop/AI/ASS-8.2(1918).py"
70.0
Pass
35.0
You've reached the limit for chat me... Upgrade
+ v ... | ☰ ×
powershell
powershell
powershell
You've reached the limit for chat me... Upgrade
+ v ... | ☰ ×
powershell
powershell
powershell
You've reached the limit for chat me... Upgrade
+ v ... | ☰ ×
powershell
powershell
powershell
Ln 157, Col 54 Spaces: 4 UFT-8 CRLF Python 3.11 (64-bit) 8:0 Go Live
SESSIONS ○ Q V U
```

```
AI ASS-8.2(1918).py ...
101 class StudentResult:
102     def calculate_average(self):
103         """
104             Calculates average of marks.
105         """
106         if len(self.marks) == 0:
107             return 0
108         return sum(self.marks) / len(self.marks)
109     def get_result(self):
110         """
111             Returns Pass if average >= 40 else Fail.
112         """
113         avg = self.calculate_average()
114         if avg >= 40:
115             return "Pass"
116         else:
117             return "Fail"
118
119 # Test Case
120 # Test Case 1: Pass Scenario
121 student1 = StudentResult()
122 student1.add_marks(60)
123 print(student1.calculate_average()) # Expected: 70.0
124 print(student1.get_result()) # Expected: Pass
125
126 student2 = StudentResult()
127 student2.add_marks(30)
128 student2.add_marks(35)
129 student2.add_marks(40)
130 print(student2.calculate_average()) # Expected: 35.0
131 print(student2.get_result()) # Expected: Fail
132
133 student3 = StudentResult()
134 print(student3.add_marks(-10)) # Expected: Error
135 print(student3.add_marks(120)) # Expected: Error
136
137 student4 = StudentResult()
138 print(student4.calculate_average()) # Expected: 0
139 print(student4.get_result()) # Expected: Fail
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\HP\Desktop\AI> & "C:\Program Files\Python311\python.exe" "C:/Users/HP/Desktop/AI/ASS-8.2(1918).py"
70.0
Pass
35.0
You've reached the limit for chat me... Upgrade
+ v ... | ☰ ×
powershell
powershell
powershell
You've reached the limit for chat me... Upgrade
+ v ... | ☰ ×
powershell
powershell
powershell
You've reached the limit for chat me... Upgrade
+ v ... | ☰ ×
powershell
powershell
powershell
Ln 157, Col 54 Spaces: 4 UFT-8 CRLF Python 3.11 (64-bit) 8:0 Go Live
SESSIONS ○ Q V U
```

```

PS C:\Users\HP\Desktop\AI> & "C:\Program Files\Python313\python.exe" "c:/Users/HP/Desktop/AI/AI_ASS_8.py"
70.0
Pass
35.0
Fail
Error: Marks must be between 0 and 100
70.0
Pass
35.0
Fail
Error: Marks must be between 0 and 100
Error: Marks must be between 0 and 100
Error: Marks must be between 0 and 100
0
Fail
Error: Marks must be between 0 and 100
Error: Marks must be between 0 and 100
Error: Marks must be between 0 and 100
0
Fail
PS C:\Users\HP\Desktop\AI>

```

Ln 157, Col 54 Spaces: 4 UTF-8 CRLF Python Chat quota reached Python 3.13 (64-bit) Go Live

Task 5 – Test-Driven Development for Username Validator

Requirements:

- Minimum length: 5 characters
- No spaces allowed
- Only alphanumeric characters

Example Test Scenarios:

```

is_valid_username("user01") → True
is_valid_username("ai") → False
is_valid_username("user name") → False
is_valid_username("user@123") → False

```

Expected Output 5

A username validation function that passes all AI-generated test cases.

```

ASS-6.5.py ASS-7.5.py ASS-7.2.py Exp8A.py AI ASS-8.2(1918).py exp-88.py
AI ASS-8.2(1918).py > ...
159 #task-5: Test-Driven Development for Username Validator
160 #requirements:
161 #. Minimum 5 characters
162 #. No spaces
163 #. Only alphanumeric characters
164 def is_valid_username(username):
165     if not isinstance(username, str):
166         return False
167     if len(username) < 5:
168         return False
169     if " " in username:
170         return False
171     if not username.isalnum():
172         return False
173     return True
174 # Test Cases
175 print(is_valid_username("user01")) # Expected: True
176 print(is_valid_username("ai")) # Expected: False
177 print(is_valid_username("user name")) # Expected: False
178 print(is_valid_username("user@123")) # Expected: False
179 print(is_valid_username("12345")) # Expected: True
180 print(is_valid_username("")) # Expected: False
181 print(is_valid_username(None)) # Expected: False

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\HP\Desktop\AI> & "C:\Program Files\Python313\python.exe" "c:/Users/HP/Desktop/AI/AI_ASS_8.py"
True
False
False
False
True
False
False
0
PS C:\Users\HP\Desktop\AI>

```

Ln 181, Col 57 Spaces: 4 UTF-8 CRLF Python Chat quota reached Python 3.13 (64-bit) Go Live

DOCTEST:

```
python -m doctest -v Ass_8.py
```

```

True
False
True
True
False
AI CODING
test
Error: Input cannot be None
6
0
0

```

```

5
Average: 70.0
Result: Pass
Average: 35.0
Result: Fail
True
False
False
False
All tests passed successfully.
Trying:
is_even(2)
Expecting:
True
ok
Trying:
is_even(7)
Expecting:
False
ok
Trying:
is_even(0)
Expecting:
True
ok
Trying:
is_even(-4)
Expecting:
True
ok
Trying:
is_even(9)
Expecting:
False
ok
10 items had no tests:
Ass_8_2
Ass_8_2.StudentResult
Ass_8_2.StudentResult.init
Ass_8_2.StudentResult.add_marks
Ass_8_2.StudentResult.calculate_average
Ass_8_2.StudentResult.get_result
Ass_8_2.is_valid_username
Ass_8_2.sum_list
Ass_8_2.to_lowercase
Ass_8_2.to_uppercase
1 items passed all tests:
5 tests in Ass_8_2.is_even
5 tests in 11 items.
5 passed and 0 failed.
Test passed.

import pytest
from Ass_8_2 import is_even, to_uppercase, to_lowercase, sum_list, StudentResult, is_vali
d_username

```

Test cases for Task 1 - Even/Odd Number Validator

```

def test_is_even():
    assert is_even(2) == True
    assert is_even(7) == False
    assert is_even(0) == True

```

```
assert is_even(-4) == True
assert is_even(9) == False
```

Test cases for Task 2 - String Case Converter

```
def test_to_uppercase():
    assert to_uppercase("ai coding") == "AI CODING"
    assert to_uppercase("") == ""
    assert to_uppercase(None) == "Error: Input cannot be None"
    assert to_uppercase("Test") == "TEST"
def test_to_lowercase():
    assert to_lowercase("TEST") == "test"
    assert to_lowercase("") == ""
    assert to_lowercase(None) == "Error: Input cannot be None"
    assert to_lowercase("Test") == "test"
```

Test cases for Task 3 - List Sum Calculator

```
def test_sum_list():
    assert sum_list([1, 2, 3]) == 6
    assert sum_list([]) == 0
    assert sum_list([-1, 5, -4]) == 0
    assert sum_list([2, "a", 3]) == 5
    assert sum_list([2, 3, -3, "a", 4]) == 6
    assert sum_list([100, -50, 20]) == 70
```

Test cases for Task 4 - StudentResult Class

```
def test_student_result():
    student = StudentResult()
    student.add_marks(60)
    student.add_marks(70)
    student.add_marks(80)
    assert student.calculate_average() == 70.0
    assert student.get_result() == "Pass"
    student = StudentResult()
    student.add_marks(30)
    student.add_marks(35)
    student.add_marks(40)
    assert student.calculate_average() == 35.0
    assert student.get_result() == "Fail"
```

Test cases for Task 5 - Username Validator

```
def test_is_valid_username():
    assert is_valid_username("user01") == True
    assert is_valid_username("ai") == False
    assert is_valid_username("user name") == False
    assert is_valid_username("user@123") == False
    assert is_valid_username("validUser") == True
    assert is_valid_username("us") == False
```

PS C:\Users\HP\Desktop\AI> & "C:\Program Files\Python313\python.exe"
"c:/Users/HP/Desktop/AI/AI ASS-8.2(1918).py"

=====

test session starts

platform win32 -- Python 3.12.3, pytest-9.0.2, pluggy-1.6.0

rootdir: PS C:\Users\HP\Desktop\AI>

A.I.AC

rootdir: PS C:\Users\HP\Desktop\AI>

A.I.AC

collected 6 items

Ass_8_2.py

[100%]

=====

6 passed in 0.09s