

ASSIGNMENT - 8.1

Name : Aravind Reddy

Hall Ticket No : 2303a51027

Batch No : 01

Course Title : AI Assisted Coding

Instructor's Name : Mr. S Naresh Kumar

Task 1: Password Strength Validator – Apply AI in Security Context

The screenshot shows a code editor window with a dark theme. The title bar says "Assignment - 8.1.py". The code itself is a Python function named `is_strong_password` that checks if a given password is strong based on length, character types (upper/lower case, digits, special characters), and spaces. It includes several test cases at the bottom. Below the code editor is a terminal window showing the execution of the script and its output.

```
18 def is_strong_password(password):
19     if len(password) < 8:
20         return False
21
22     has_upper = False
23     has_lower = False
24     has_digit = False
25     has_special = False
26     special_characters = "!@#$%^&*()--+"
27
28     for char in password:
29         if char.isupper():
30             has_upper = True
31         elif char.islower():
32             has_lower = True
33         elif char.isdigit():
34             has_digit = True
35         elif char in special_characters:
36             has_special = True
37         elif char == " ":
38             return False
39
40     return has_upper and has_lower and has_digit and has_special
41
42 print(is_strong_password("Abcd@123")) # Should return True
43 print(is_strong_password("abcd123")) # Should return False
44 print(is_strong_password("Abcd123!")) # Should return True
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER

```
/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding/Assignments/Assignment - 8.1.py"
● (base) + AI Assissted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding/Assignments/Assi
True
False
True
◆ (base) + AI Assissted Coding
```

Task 2: Number Classification with Loops – Apply AI for Edge Case Handling

Welcome

Assignment - 8.1.py X

Assignments > Assignment - 8.1.py > ...

```
64 def classify_number(n):
65     if n is None or isinstance(n, str):
66         return "Invalid input"
67
68     if n > 0:
69         return "Positive"
70     elif n < 0:
71         return "Negative"
72     else:
73         return "Zero"
74
75 # Assert test cases
76 assert classify_number(10) == "Positive"
77 assert classify_number(-5) == "Negative"
78 assert classify_number(0) == "Zero"
79 assert classify_number("string") == "Invalid input"
80 assert classify_number(None) == "Invalid input"
81 print(classify_number(10)) # Should return "Positive"
82 print(classify_number(-5)) # Should return "Negative"
83 print(classify_number(0)) # Should return "Zero"
84 print(classify_number("string")) # Should return "Invalid input"
85 print(classify_number(None)) # Should return "Invalid input"
86
87 print("All assert tests passed successfully!")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER

```
/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding/Assignments/Assignment - 8.1.py"
● (base) → AI Assissted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding/Ass
Positive
Negative
Zero
Invalid input
Invalid input
All assert tests passed successfully!
❖ (base) → AI Assissted Coding
```

Task 3 : Anagram Checker – Apply AI for String Analysis

Welcome

Assignment - 8.1.py X

Assignments > Assignment - 8.1.py > ...

```
103
104     import string
105     def is_anagram(str1, str2):
106         # Remove spaces and punctuation, and convert to lowercase
107         translator = str.maketrans('', '', string.punctuation + ' ')
108         str1_cleaned = str1.translate(translator).lower()
109         str2_cleaned = str2.translate(translator).lower()
110         # Sort the characters of both strings and compare
111         return sorted(str1_cleaned) == sorted(str2_cleaned)
112
# Assert test cases
113 assert is_anagram("listen", "silent") == True
114 assert is_anagram("hello", "world") == False
115 assert is_anagram("Dormitory", "Dirty Room") == True
116 assert is_anagram("", "") == True # Edge case: empty strings
117 assert is_anagram("abc", "abc") == True # Edge case: identical words
118 print(is_anagram("listen", "silent")) # Should return True
119 print(is_anagram("hello", "world")) # Should return False
120 print(is_anagram("Dormitory", "Dirty Room")) # Should return True
121 print(is_anagram("", "")) # Should return True
122 print(is_anagram("abc", "abc")) # Should return True
123 print(["All assert tests passed successfully!"])
124
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER

```
/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding/Assignments/Assignment - 8.1.py"
● (base) → AI Assissted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding/Ass:
True
False
True
True
True
True
All assert tests passed successfully!
❖ (base) → AI Assissted Coding
```

Task 4 : Inventory Class – Apply AI to Simulate Real- World Inventory System

The screenshot shows a code editor interface with a dark theme. At the top, there's a navigation bar with 'Welcome' and 'Assignment - 8.1.py'. Below it, a breadcrumb trail says 'Assignments > Assignment - 8.1.py > ...'. The main area contains the following Python code:

```
147 class Inventory:
148     def __init__(self):
149         self.stock = {}
150
151     def add_item(self, name, quantity):
152         if name in self.stock:
153             self.stock[name] += quantity
154         else:
155             self.stock[name] = quantity
156
157     def remove_item(self, name, quantity):
158         if name in self.stock and self.stock[name] >= quantity:
159             self.stock[name] -= quantity
160         else:
161             raise ValueError("Not enough stock to remove")
162
163     def get_stock(self, name):
164         return self.stock.get(name, 0)
165
166 # Assert test cases
167 inv = Inventory()
168 inv.add_item("Pen", 10)
169 assert inv.get_stock("Pen") == 10
170 inv.remove_item("Pen", 5)
171 assert inv.get_stock("Pen") == 5
172 inv.add_item("Book", 3)
173 assert inv.get_stock("Book") == 3
174 print(inv.get_stock("Pen")) # Should return 5
175 print(inv.get_stock("Book")) # Should return 3
176 print("All assert tests passed successfully!")
```

Below the code, there's a toolbar with buttons for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected), 'PORTS', and 'SPELL CHECKER'. Underneath the toolbar, the terminal output shows:

```
/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment  
● (base) + AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted  
3  
All assert tests passed successfully!
```

Task 5 : Date Validation & Formatting – Apply AI for Data Validation

Welcome

Assignment - 8.1.py

Assignments > Assignment - 8.1.py > ...

```
197
198     from datetime import datetime
199     def validate_and_format_date(date_str):
200         try:
201             date_obj = datetime.strptime(date_str, "%m/%d/%Y")
202             return date_obj.strftime("%Y-%m-%d")
203         except ValueError:
204             return "Invalid Date"
205     # Assert test cases
206     assert validate_and_format_date("10/15/2023") == "2023-10-15"
207     assert validate_and_format_date("02/30/2023") == "Invalid Date"
208     assert validate_and_format_date("01/01/2024") == "2024-01-01"
209     assert validate_and_format_date("13/01/2023") == "Invalid Date" # Invalid month
210     assert validate_and_format_date("00/10/2023") == "Invalid Date" # Invalid month
211     assert validate_and_format_date("12/32/2023") == "Invalid Date" # Invalid day
212     print(validate_and_format_date("10/15/2023")) # Should return "2023-10-15"
213     print(validate_and_format_date("02/30/2023")) # Should return "Invalid Date"
214     print(validate_and_format_date("01/01/2024")) # Should return "2024-01-01"
215     print(validate_and_format_date("00/10/2023")) # Should return "Invalid Date"
216     print(validate_and_format_date("12/32/2023")) # Should return "Invalid Date"
217     print("All assert tests passed successfully!")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER

```
/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding/Assignments/Assignment - 8.1.
● (base) → AI Assissted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding
2023-10-15
Invalid Date
2024-01-01
Invalid Date
Invalid Date
All assert tests passed successfully!
❖ (base) → AI Assissted Coding
```