

2303a51739

Lab assignment 8.3

Task 1: Email Validation using TDD

Scenario

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

The screenshot shows a Python file named `ai_lab 8.3.py` open in the editor. The code implements an `is_valid_email` function and contains several test cases for validating emails. The terminal at the bottom shows the execution of the script and its output.

```
File Ed Selection View Go Run ... < > Q AI Coding ai lab 8.3.py Welcome assignment 3.py AI lab 43.py lab assignment 5.4.py lab assignment 6.3.py lab assignment 7.3.py target_file.py ai exam q1.py ai exam q2.py ai lab 10.3.txt ai lab 8.3.py

1 import re
2 import unittest
3
4 # Function Implementation
5 def is_valid_email(email: str) -> bool:
6     if not isinstance(email, str):
7         return False
8
9     if email.count('@') != 1:
10        return False
11
12     if "@" not in email or "." not in email:
13        return False
14
15     if email[0] in ".@_" or email[-1] in ".@_":
16        return False
17
18     pattern = r'^[A-Za-z0-9][A-Za-z0-9_.]*@[A-Za-z0-9]+\.[A-Za-z]{2,}$'
19     return re.match(pattern, email) is not None
20
21
22 # Test Cases
23 class TestEmailValidation(unittest.TestCase):
24
25     def test_valid_email(self):
26         self.assertTrue(is_valid_email("test@example.com"))
27         self.assertTrue(is_valid_email("user.name@gmail.com"))
28
29     def test_invalid_email(self):
30         self.assertFalse(is_valid_email("testexample.com"))
31         self.assertFalse(is_valid_email("test@example.com"))
32         self.assertFalse(is_valid_email("examnle.com"))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ...
```

Ran 2 tests in 0.001s

OK

PS D:\AI Coding> & C:/Users/AMOALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/ai_lab 8.3.py"

Ran 2 tests in 0.002s

Task 2: Grade Assignment using Loops

Scenario

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

The screenshot shows the VS Code interface with two tabs open: `ai lab 8.3.py` and `ai lab 8.3.py`. The code implements a `grade_assignment` function that takes a score and returns a grade based on a switch statement. It also includes a `TestGradeAssignment` class with several test cases using `assertEqual` to check if the function returns the correct grade for different scores. The terminal below shows the execution of the script and the results of the unit tests.

```
1 import unittest
2
3 # Function Implementation
4 def assign_grade(score):
5     if not isinstance(score, (int, float)):
6         raise TypeError("Score must be numeric")
7
8     if score < 0 or score > 100:
9         raise ValueError("Score must be between 0 and 100")
10
11    if 90 <= score <= 100:
12        return "A"
13    elif 80 <= score <= 89:
14        return "B"
15    elif 70 <= score <= 79:
16        return "C"
17    elif 60 <= score <= 69:
18        return "D"
19    else:
20        return "F"
21
22
23 # Test Cases
24 class TestGradeAssignment(unittest.TestCase):
25
26     def test_valid_grades(self):
27         self.assertEqual(assign_grade(95), "A")
28         self.assertEqual(assign_grade(85), "B")
29         self.assertEqual(assign_grade(75), "C")
30         self.assertEqual(assign_grade(65), "D")
31         self.assertEqual(assign_grade(50), "F")
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50 if __name__ == "__main__":
51     unittest.main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

...

Ran 2 tests in 0.002s

OK
PS D:\AI Coding & C:/Users/NOJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/ai lab 8.3.py"
...

Extension Bisect is active and has disabled 3 extensions. You can still reproduce the problem and proceed by selecting these options.
I can't reproduce I can reproduce

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

...

Ran 3 tests in 0.001s

OK
PS D:\AI Coding>

Extension Bisect is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options.
I can't reproduce I can reproduce Stop Bisect

Ln 52, Col 1 Spaces: 4 - UTF-8 CR/LF [Python 3.13]

Task 3: Sentence Palindrome Checker

Scenario

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

The screenshot shows a code editor window with the following Python code:

```
1 # ai lab 8.3.py > ...
2 import re
3 import unittest
4
5 # Function Implementation
6 def is_sentence_palindrome(sentence: str) -> bool:
7     if not isinstance(sentence, str):
8         return False
9
10    cleaned = re.sub(r'[^A-Za-z0-9]', ' ', sentence).lower()
11    return cleaned == cleaned[::-1]
12
13 # Test Cases
14 class TestSentencePalindrome(unittest.TestCase):
15
16    def test_palindrome_sentences(self):
17        self.assertTrue(is_sentence_palindrome("A man a plan a canal Panama"))
18        self.assertTrue(is_sentence_palindrome("Madam"))
19
20    def test_non_palindrome_sentences(self):
21        self.assertFalse(is_sentence_palindrome("Hello World"))
22        self.assertFalse(is_sentence_palindrome("Python Programming"))
23
24
25 if __name__ == "__main__":
26     unittest.main()
```

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

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... | x
...
Ran 3 tests in 0.001s
OK
PS D:\AI Coding> & C:/Users/ANDALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/ai lab 8.3.py"
...
Ran 2 tests in 0.001s
OK
PS D:\AI Coding>
```

A status bar at the bottom indicates: Line 32 Col 1 Status bar: Line 32 Col 1 Python 3.13.0

A message in the terminal says: Extension Biject is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options.

Task 4: ShoppingCart Class

Scenario

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

The screenshot shows the VS Code interface with the following code:

```
1 #!/usr/bin/python
2
3 # Class Implementation
4 class ShoppingCart:
5
6     def __init__(self):
7         self.items = {}
8
9     def add_item(self, name, price):
10        if price < 0:
11            raise ValueError("Price cannot be negative")
12        self.items[name] = price
13
14    def remove_item(self, name):
15        if name in self.items:
16            del self.items[name]
17        else:
18            raise KeyError("Item not found")
19
20    def total_cost(self):
21        return sum(self.items.values())
22
23
24 # Test Cases
25 class TestShoppingCart(unittest.TestCase):
26
27     def setup(self):
28         self.cart = ShoppingCart()
29
30     def test_add_item(self):
31         self.cart.add_item("Book", 200)
32         self.assertEqual(self.cart.total_cost(), 200)
33
34     def test_remove_item(self):
35         self.cart.add_item("Pen", 50)
36         self.cart.remove_item("Pen")
37         self.assertEqual(self.cart.total_cost(), 0)
38
39     def test_empty_cart(self):
40         self.assertEqual(self.cart.total_cost(), 0)
41
42     def test_remove_nonexistent_item(self):
43         with self.assertRaises(KeyError):
44             self.cart.remove_item("Laptop")
45
46
47 if __name__ == "__main__":
48     unittest.main()
```

The terminal output shows the tests running successfully:

```
Ran 2 tests in 0.001s
OK
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/ai lab 8.3.py"
...
Ran 4 tests in 0.001s
OK
PS D:\AI Coding>
```

A tooltip in the bottom right corner indicates that the Extension Biscet is active and has disabled 3 extensions.

The screenshot shows the VS Code interface with the following code:

```
1 #!/usr/bin/python
2
3 class TestShoppingCart(unittest.TestCase):
4
5     def setup(self):
6         self.cart = ShoppingCart()
7
8     def test_add_item(self):
9         self.cart.add_item("Book", 200)
10        self.assertEqual(self.cart.total_cost(), 200)
11
12     def test_remove_item(self):
13         self.cart.add_item("Pen", 50)
14         self.cart.remove_item("Pen")
15         self.assertEqual(self.cart.total_cost(), 0)
16
17     def test_empty_cart(self):
18         self.assertEqual(self.cart.total_cost(), 0)
19
20     def test_remove_nonexistent_item(self):
21         with self.assertRaises(KeyError):
22             self.cart.remove_item("Laptop")
23
24
25 if __name__ == "__main__":
26     unittest.main()
```

The terminal output shows the tests running successfully:

```
Ran 2 tests in 0.001s
OK
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/ai lab 8.3.py"
...
Ran 4 tests in 0.001s
OK
PS D:\AI Coding>
```

A tooltip in the bottom right corner indicates that the Extension Biscet is active and has disabled 3 extensions.

Task 5: Date Format Conversion

Scenario

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

A screenshot of the Visual Studio Code (VS Code) interface. The main editor window contains Python code for a date conversion utility. The code includes a function `convert_date_format` that checks if the input is a string and matches it against a YYYY-MM-DD pattern. It also includes test cases for valid dates, invalid formats, and invalid types. The terminal below shows the command `python ai_lab_8.3.py` being run, followed by the output of running 4 tests in 0.001s and the command `ps` being run again. A status bar at the bottom shows the file is 211 words long and was last modified on 2/18/2026 at 9:44 AM.

```
File Edit Selection View Go Run ... ← → Q AI Coding
ai lab 8.3.py > ...
1 import re
2 import unittest
3
4 # Function Implementation
5 def convert_date_format(date_str):
6     if not isinstance(date_str, str):
7         raise TypeError("Date must be a string")
8
9     pattern = r"\d{4}-\d{2}-\d{2}"
10    if not re.match(pattern, date_str):
11        raise ValueError("Invalid date format. Use YYYY-MM-DD")
12
13    year, month, day = date_str.split("-")
14    return f"{day}-{month}-{year}"
15
16
17 # Test Cases
18 class TestDataConversion(unittest.TestCase):
19
20     def test_valid_date(self):
21         self.assertEqual(convert_date_format("2023-10-15"), "15-10-2023")
22
23     def test_invalid_format(self):
24         with self.assertRaises(ValueError):
25             convert_date_format("15-10-2023")
26
27     def test_invalid_type(self):
28         with self.assertRaises(TypeError):
29             convert_date_format(20231015)
30
31
32 if __name__ == "__main__":
33     unittest.main()
34
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
+ v ... | □ x
powershell Python Python Python Python
Extension Bisection is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options.
I can't reproduce I can reproduce Stop Bisection
Ran 4 tests in 0.001s
OK
PS D:\AI Coding & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/ai_lab_8.3.py"
...
Ran 3 tests in 0.001s
OK
PS D:\AI Coding

```

A screenshot of the Visual Studio Code (VS Code) interface. The main editor window contains Python code for a date conversion utility. The code includes a function `convert_date_format` that checks if the input is a string and matches it against a YYYY-MM-DD pattern. It also includes test cases for valid dates, invalid formats, and invalid types. The terminal below shows the command `python ai_lab_8.3.py` being run, followed by the output of running 4 tests in 0.001s and the command `ps` being run again. A status bar at the bottom shows the file is 211 words long and was last modified on 2/18/2026 at 9:44 AM.

```
File Edit Selection View Go Run ... ← → Q AI Coding
ai lab 8.3.py > ...
5 def convert_date_format(date_str):
6     raise ValueError("Invalid date format. Use YYYY-MM-DD")
7
8     year, month, day = date_str.split("-")
9     return f"{day}-{month}-{year}"
10
11
12 # Test Cases
13 class TestDataConversion(unittest.TestCase):
14
15     def test_valid_date(self):
16         self.assertEqual(convert_date_format("2023-10-15"), "15-10-2023")
17
18     def test_invalid_format(self):
19         with self.assertRaises(ValueError):
20             convert_date_format("15-10-2023")
21
22     def test_invalid_type(self):
23         with self.assertRaises(TypeError):
24             convert_date_format(20231015)
25
26
27 if __name__ == "__main__":
28     unittest.main()
29
30
31
32
33
34
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
+ v ... | □ x
powershell Python Python Python Python
Extension Bisection is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options.
I can't reproduce I can reproduce Stop Bisection
Ran 4 tests in 0.001s
OK
PS D:\AI Coding & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/ai_lab_8.3.py"
...
Ran 3 tests in 0.001s
OK
PS D:\AI Coding

```

A screenshot of the Visual Studio Code (VS Code) interface. The main editor window contains Python code for a date conversion utility. The code includes a function `convert_date_format` that checks if the input is a string and matches it against a YYYY-MM-DD pattern. It also includes test cases for valid dates, invalid formats, and invalid types. The terminal below shows the command `python ai_lab_8.3.py` being run, followed by the output of running 4 tests in 0.001s and the command `ps` being run again. A status bar at the bottom shows the file is 211 words long and was last modified on 2/18/2026 at 9:44 AM.

```
File Edit Selection View Go Run ... ← → Q AI Coding
ai lab 8.3.py > ...
...
Ran 4 tests in 0.001s
OK
PS D:\AI Coding & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/ai_lab_8.3.py"
...
Ran 3 tests in 0.001s
OK
PS D:\AI Coding
22°C
Sunny
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
+ v ... | □ x
powershell Python Python Python Python
Extension Bisection is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options.
I can't reproduce I can reproduce Stop Bisection
In 34, Col 1  Spaces: 4  UTR-8  CRLF  Python  3.13.7
ENG IN 9:44 AM 2/18/2026
```

