

2303a51504

## Lab assignment 8.3

### Task 1: Email Validation using TDD

#### Scenario

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

```
File Edit Selection View Go Run ... ← → Q AI Coding
ai lab 8.3.py > TestEmailValidation > test_invalid_email
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("user.name@.com"))
33
34
35     if __name__ == "__main__":
36         unittest.main()
37
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
...
Ran 2 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 0.002s
OK
PS D:\AI Coding>
```

```
File Edit Selection View Go Run ... ← → Q AI Coding
ai lab 8.3.py > TestEmailValidation > test_invalid_email
5 def is_valid_email(email: str) -> bool:
6     return False
7
8     pattern = r'^[A-Za-z0-9][A-Za-z0-9-]*@[A-Za-z0-9]+\.[A-Za-z]{2,}$'
9     return re.match(pattern, email) is not None
10
11
12 # Test Cases
13 class TestEmailValidation(unittest.TestCase):
14
15     def test_valid_email(self):
16         self.assertTrue(is_valid_email("test@example.com"))
17         self.assertTrue(is_valid_email("user.name@gmail.com"))
18
19     def test_invalid_email(self):
20         self.assertFalse(is_valid_email("testexample.com"))
21         self.assertFalse(is_valid_email("test@example.com"))
22         self.assertFalse(is_valid_email("user.name@.com"))
23         self.assertFalse(is_valid_email("test@com"))
24
25
26     if __name__ == "__main__":
27         unittest.main()
28
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
...
Ran 2 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 0.002s
OK
PS D:\AI Coding>
```

## 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 is as follows:

```
1 # Function Implementation
2 def assign_grade(score):
3     if not isinstance(score, (int, float)):
4         raise TypeError("Score must be numeric")
5
6     if score < 0 or score > 100:
7         raise ValueError("Score must be between 0 and 100")
8
9     if 90 <= score <= 100:
10        return "A"
11    elif 80 <= score <= 89:
12        return "B"
13    elif 70 <= score <= 79:
14        return "C"
15    elif 60 <= score <= 69:
16        return "D"
17    else:
18        return "F"
19
20
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()
```

The terminal output shows the execution of the tests:

```
PS D:\AI Coding & C:/Users/NOJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/ai lab 8.3.py"
...
Ran 2 tests in 0.002s
OK
PS D:\AI Coding>
PS D:\AI Coding>
```

A status bar at the bottom right indicates: Extension Bisect is active and has disabled 3 extensions. You can still reproduce the problem and proceed by selecting from these options. Buttons for 'I can't reproduce' and 'I can reproduce' are also present.

The second screenshot shows the same code structure, but the terminal output is different:

```
PS D:\AI Coding & C:/Users/NOJALI/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>
PS D:\AI Coding>
```

A status bar at the bottom right indicates: Extension Bisect is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options. Buttons for 'I can't reproduce' and 'I can reproduce' are also present.

### 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:

```
ai lab 8.3.py > ...
1 import re
2 import unittest
3
4 # Function Implementation
5 def is_sentence_palindrome(sentence: str) -> bool:
6     if not isinstance(sentence, str):
7         return False
8
9     cleaned = re.sub(r'[^\w\s]', '', sentence).lower()
10    return cleaned == cleaned[::-1]
11
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 3.11.4

A tooltip message in the terminal area states: Extension Biject is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options. Buttons: I can't reproduce, I can reproduce, Stop Biject.

## 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:

```
PS D:\AI Coding> & C:/Users/ANJALI/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> & 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 says: "Extension Biscet is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options." Buttons for "I can't reproduce", "I can reproduce", and "Stop Biscet" are shown.

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:

```
PS D:\AI Coding> & C:/Users/ANJALI/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> & 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 says: "Extension Biscet is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options." Buttons for "I can't reproduce", "I can reproduce", and "Stop Biscet" are shown.

## Task 5: Date Format Conversion

### Scenario

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

The screenshot shows a VS Code interface with the following code in the editor:

```
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 TestDateConversion(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()
```

The terminal below the editor shows the output of running the tests:

```
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 status bar at the bottom right indicates "Extension Bisection is active and has disabled 3 extensions. Check if you can still reproduce the problem and proceed by selecting from these options." with buttons for "I can't reproduce", "I can reproduce", and "Stop Bisection".

The screenshot shows a second instance of VS Code with the same code and test results as the first one. The terminal output is identical:

```
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
```

The status bar at the bottom right also displays the message about Extension Bisection being active.

