

# COS80022 – Software Quality and Testing

## Test Report for Lab 7

Student Name: Arun Ragavendhar Arunachalam Palaniyappan  
Student ID: 104837257  
Student email: 104837257@student.swin.edu.au  
Submission date: 21/04/2024

## 1. Equivalence Partitioning

### 1.1 Test cases

By the given system definition,

Customer gets an “appetizer” on reaching exactly 500 points, gets a “main course” on reaching 1500 points and a “fine dining set” when they exceed 5000 points and maximum point limit reaching 10,000 points.

- ⇒ Input -  $<0$  – Partition 1 – “invalid”
- ⇒ Input - 0-499 - Partition 2 – “no reward”
- ⇒ Input - 500-1499 - Partition 3 – “appetizer”
- ⇒ Input - 1500 – 5000 – Partition 4 – “main course”
- ⇒ Input – 5001-10000 – Partition 5 – “fine dining set”
- ⇒ Input -  $>10000$  – Partition 6 – “unable to earn  $>10000$ ”

So, we have 6 partitions here, for each partition let us pick 1 test input value within the partition to test the function.

TC ID	Input for each Partition	Input Data	Expected Result	Actual Ouput
TC01	Integer $<0$	-45	“invalid”	“invalid”
TC02	Integer (0-499)	300	“no reward”	“no reward”
TC03	Integer (500-1499)	800	“appetizer”	“appetizer”
TC04	Integer (1500-5000)	3500	“main course”	“main course”
TC05	Integer (5001-10000)	8000	“fine dining set”	“fine dining set”
TC06	Integer $>10000$	12000	“unable to earn $>10000$ ”	“unable to earn $>10000$ ”

### 1.2 Screenshot of test results

```

PS C:\Users\arunr\Videos\SQT_lab7_pack2\SQT_lab7_pack2> pytest test_lab7_EP_104837257.py
===== test session starts =====
platform win32 -- Python 3.11.9, pytest-8.1.1, pluggy-1.4.0
rootdir: C:\Users\arunr\Videos\SQT_lab7_pack2\SQT_lab7_pack2
collected 6 items

test_lab7_EP_104837257.py ..... [100%]

===== 6 passed in 0.03s =====
PS C:\Users\arunr\Videos\SQT_lab7_pack2\SQT_lab7_pack2> |

```

The given program **loyalty.py** successfully passes all the test cases given , for the equivalence partition testing . Let us further test the program using boundary value analysis.

## 2. Boundary Value Analysis

### 2.1 Test cases

#### 2 value Boundary value sets

(-1,0)

(499,500)

(1499,1500)

(5000,5001)

(10000,10001)

#### 3 value Boundary value sets

(-1,0,1)

(499,500,501)

(1499,1500,1501)

(5000,5001,5002)

(10000,10001,10002)

#### 2 BV:

TC ID	Boundary/Partition Tested	Input Data	Expected Result	Actual Output
TC01	Integer<0	-1	"invalid"	"invalid"
TC02	Integer(0-499)	0	"no reward"	"no reward"
TC03	Integer(0-499)	499	"no reward"	"no reward"
TC04	Integer(500-1499)	500	"appetizer"	"appetizer"
TC05	Integer(500-1499)	1499	"appetizer"	"appetizer"
TC06	Integer(1500-5000)	1500	"main course"	"main course"
<b>TC07</b>	<b>Integer(1500-5000)</b>	<b>5000</b>	<b>"main course"</b>	<b>"fine dining set"</b>
TC08	Integer(5001-10000)	5001	"fine dining set"	"fine dining set"
TC09	Integer(5001-10000)	10000	"fine dining set"	"fine dining set"
TC10	Integer(>10000)	10001	"unable to earn>10000"	"unable to earn>10000"

#### 3 BV:

TC NO	Boundary/Partition Tested	Input Data	Expected Result	Actual Output
TC01	Integer<0	-1	"invalid"	"invalid"
TC02	Integer(0-499)	0	"no reward"	"no reward"
TC03	Integer(0-499)	1	"no reward"	"no reward"
TC04	Integer(0-499)	499	"no reward"	"no reward"
TC05	Integer(500-1499)	500	"appetizer"	"appetizer"
TC06	Integer(500-1499)	501	"appetizer"	"appetizer"
TC07	Integer(500-1499)	1499	"appetizer"	"appetizer"
TC08	Integer(1500-5000)	1500	"main course"	"main course"
TC09	Integer(1500-5000)	1501	"main course"	"main course"
<b>TC10</b>	<b>Integer(1500-5000)</b>	<b>5000</b>	<b>"main course"</b>	<b>"fine dining set"</b>
TC11	Integer(5001-10000)	5001	"fine dining set"	"fine dining set"
TC12	Integer(5001-10000)	5002	"fine dining set"	"fine dining set"
TC13	Integer(5001-10000)	10000	"fine dining set"	"fine dining set"
TC14	Integer(5001-10000)	10001	"unable to earn>10000"	"unable to earn>10000"
TC15	Integer(>10000)	10002	"unable to earn>10000"	"unable to earn>10000"

## 2.2 Screenshot of test results

### OUTPUT OF 2 BV ANALYSIS:

- Test result on running 2 BV analysis testing.
- One failed case, where “5000” should have actually been “main course “but is wrongly been displayed as “fine dining set.”

```
PS C:\Users\arunr\Videos\SQT_lab7_pack2\SQT_lab7_pack2> pytest test_lab7_BVA_104837257.py
===== test session starts =====
platform win32 -- Python 3.11.9, pytest-8.1.1, pluggy-1.4.0
rootdir: C:\Users\arunr\Videos\SQT_lab7_pack2\SQT_lab7_pack2
collected 10 items

test_lab7_BVA_104837257.py .....F... [100%]

===== FAILURES =====
test_maincourse2
-----
def test_maincourse2():
    #The following "3500" is the input data for the test case
    points = 5000
    #The following executes the program with the input data, and get result.
    result = loyalty.decideReward(points)
    #The following verifies if the result is consistent with the expectation.
    assert result == 'main course'
>
E       AssertionError: assert 'fine dining set' == 'main course'
E
E       - main course
E       + fine dining set
test_lab7_BVA_104837257.py:63: AssertionError
----- Captured stdout call -----
The reward is: fine dining set
===== short test summary info =====
FAILED test_lab7_BVA_104837257.py::test_maincourse2 - AssertionError: assert 'fine dining set' == 'main course'
===== 1 failed, 9 passed in 0.08s =====
PS C:\Users\arunr\Videos\SQT_lab7_pack2\SQT_lab7_pack2>
```

### OUTPUT OF 3 BV ANALYSIS:

- Test result on running 3 BV analysis testing.
- Again, the same test case failed TC, where “5000” should have actually been “main course “but is wrongly been displayed as “fine dining set.”

```
PS C:\Users\arunr\Videos\SQT_lab7_pack2\SQT_lab7_pack2> pytest test_lab7_BVA_104837257.py
===== test session starts =====
platform win32 -- Python 3.11.9, pytest-8.1.1, pluggy-1.4.0
rootdir: C:\Users\arunr\Videos\SQT_lab7_pack2\SQT_lab7_pack2
collected 15 items

test_lab7_BVA_104837257.py .....F..... [100%]

===== FAILURES =====
test_maincourse3
-----
def test_maincourse3():
    #The following "3500" is the input data for the test case
    points = 5000
    #The following executes the program with the input data, and get result.
    result = loyalty.decideReward(points)
    #The following verifies if the result is consistent with the expectation.
    assert result == 'main course'
>
E       AssertionError: assert 'fine dining set' == 'main course'
E
E       - main course
E       + fine dining set
test_lab7_BVA_104837257.py:91: AssertionError
----- Captured stdout call -----
The reward is: fine dining set
===== short test summary info =====
FAILED test_lab7_BVA_104837257.py::test_maincourse3 - AssertionError: assert 'fine dining set' == 'main course'
===== 1 failed, 14 passed in 0.11s =====
PS C:\Users\arunr\Videos\SQT_lab7_pack2\SQT_lab7_pack2>
```

### 3. Analysis of test results

The given program has an error in the boundary condition of “main course”

**Current output:**

Reward: 5000

Output: fine dining set

Reward: 5001

Output: fine dining set

**Actual expected output as per program deifinition:**

Reward: 5000

Output: main course

Reward: 5001

Output: fine dining set

**Part of the program that has the error:**

```
else:
    if points < 5000:
        decision = "main course"
    else:
        if points <= 10000:
            decision = "fine dining set"
        else:
            decision = "unable to earn > 10000"
print("The reward is: " + decision)
return decision
```

**De-bugged program after code correction:**

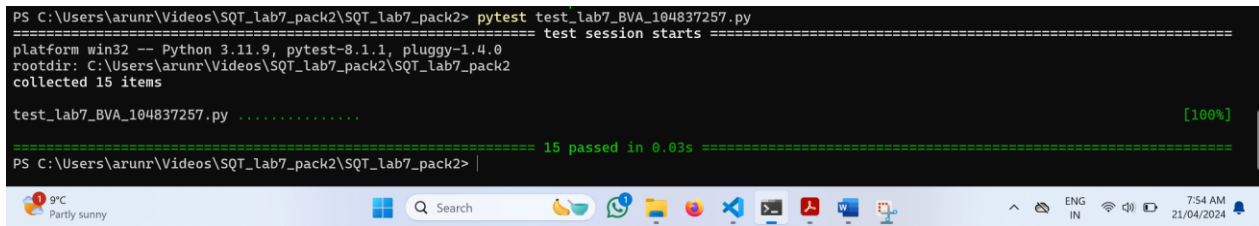
```
else:
    if points <= 5000:
        decision = "main course"
    else:
        if points <= 10000:
            decision = "fine dining set"
        else:
            decision = "unable to earn > 10000"
print("The reward is: " + decision)
return decision
```

## Test Output after Code correction:

```
PS C:\Users\arunr\Videos\SQT_lab7_pack2\SQT_lab7_pack2> pytest test_lab7_BVA_104837257.py
===== test session starts =====
platform win32 -- Python 3.11.9, pytest-8.1.1, pluggy-1.4.0
rootdir: C:\Users\arunr\Videos\SQT_lab7_pack2\SQT_lab7_pack2
collected 15 items

test_lab7_BVA_104837257.py ..... [100%]

===== 15 passed in 0.03s =====
PS C:\Users\arunr\Videos\SQT_lab7_pack2\SQT_lab7_pack2> |
```



- Now all the actual output match the expected output and all test cases have passed.
- Hence testing successfully completed.