



In-class Exercises

Input Space Partition Testing

Software Testing & Maintenance

SWE 437

<http://go.gmu.edu/swe437>

Dr. Brittany Johnson-Matthews

(Dr. B for short)

In-class Exercise #1: Test Criteria

Consider the following abstract IDM:

Characteristic	Block 1	Block 2	Block 3	Block 4	Block 5
A	a1	a2	a3	a4	a5
B	b1	b2	b3		
C	c1	c2	c3	c4	c5
D	d1	d2			

1. How many tests would we need to satisfy ACoC?
2. How many tests are needed to satisfy ECC?
3. Write abstract tests to satisfy ECC.
4. How many tests are needed to satisfy BCC?
5. Assume the base values are a1, b1, c1, and d1. Write abstract tests to satisfy BC.
6. How many tests would we need to satisfy MBCC?

In-class Exercise #1: Test Criteria

Characteristic	Block 1	Block 2	Block 3	Block 4	Block 5
A	a1	a2	a3	a4	a5
B	b1	b2	b3		
C	c1	c2	c3	c4	c5
D	d1	d2			

1. How many tests would we need to satisfy ACoC? $5*3*5*2 = \mathbf{150 \text{ tests}}$

In-class Exercise #1: Test Criteria

Characteristic	Block 1	Block 2	Block 3	Block 4	Block 5
A	a1	a2	a3	a4	a5
B	b1	b2	b3		
C	c1	c2	c3	c4	c5
D	d1	d2			

1. How many tests would we need to satisfy ACoC? $5*3*5*2 =$ **150 tests**
2. How many tests are needed to satisfy ECC? Max # blocks 5 = **minimum of 5 tests**

In-class Exercise #1: Test Criteria

Characteristic	Block 1	Block 2	Block 3	Block 4	Block 5
A	a1	a2	a3	a4	a5
B	b1	b2	b3		
C	c1	c2	c3	c4	c5
D	d1	d2			

1. How many tests would we need to satisfy ACoC? $5 \times 3 \times 5 \times 2 =$ **150 tests**
2. How many tests are needed to satisfy ECC? Max # blocks 5 = **minimum of 5 tests**
3. Write abstract tests to satisfy ECC. **(a1, b1, c1, d1) (a2, b2, c2, d2) (a3, b3, c3, d1) (a4, b3, c4, d2) (a5, b2, c5, d1)**

In-class Exercise #1: Test Criteria

Characteristic	Block 1	Block 2	Block 3	Block 4	Block 5
A	a1	a2	a3	a4	a5
B	b1	b2	b3		
C	c1	c2	c3	c4	c5
D	d1	d2			

1. How many tests would we need to satisfy ACoC? $5*3*5*2 = \mathbf{150 \text{ tests}}$
2. How many tests are needed to satisfy ECC? Max # blocks 5 = **minimum of 5 tests**
3. Write abstract tests to satisfy ECC. **(a1, b1, c1, d1) (a2, b2, c2, d2) (a3, b3, c3, d1) (a4, b3, c4, d2) (a5, b2, c5, d1)**
4. How many tests are needed to satisfy BCC? $1+4+2+4+1 = \mathbf{12 \text{ tests}}$

In-class Exercise #1: Test Criteria

Characteristic	Block 1	Block 2	Block 3	Block 4	Block 5
A	a1	a2	a3	a4	a5
B	b1	b2	b3		
C	c1	c2	c3	c4	c5
D	d1	d2			

5. Assume the base values are a1, b1, c1, and d1. Write abstract tests to satisfy BC.

Base test = **(a1, b1, c1, d1)**

Tests = **(a2, b1, c1, d1)**; **(a3, b1, c1, d1)**; **(a4, b1, c1, d1)**; **(a5, b1, c1, d1)**; (a1, **b2, c1, d1**), (a1, **b3, c1, d1**); (a1, b1, **c2, d1**); (a1, b1, **c3, d1**); (a1, b1, **c4, d1**); (a1, b1, **c5, d1**); (a1, b1, c1, **d2**)

In-class Exercise #1: Test Criteria

Characteristic	Block 1	Block 2	Block 3	Block 4	Block 5
A	a1	a2	a3	a4	a5
B	b1	b2	b3		
C	c1	c2	c3	c4	c5
D	d1	d2			

5. Assume the base values are a1, b1, c1, and d1. Write abstract tests to satisfy BC.

Base test = **(a1, b1, c1, d1)**

Tests = **(a2, b1, c1, d1)**; **(a3, b1, c1, d1)**; **(a4, b1, c1, d1)**; **(a5, b1, c1, d1)**; (a1, **b2, c1, d1**), (a1, **b3, c1, d1**); (a1, b1, **c2, d1**); (a1, b1, **c3, d1**); (a1, b1, **c4, d1**); (a1, b1, **c5, d1**); (a1, b1, c1, **d2**)

6. How many tests would we need to satisfy MBCC? $2 + 2*3 + 2*1 + 2*3 + 2*0 =$ **24 tests**

In-class Exercise #2: ISP Tests

Design ISP tests for the simple web application [calculate](#).

Design the input domain model, abstract tests, and test values. Use the each choice criterion (ECC).

In your group, go through the following steps:

1. List the testable function or functions.
2. List the inputs.
3. Define characteristics for the inputs.
4. Define blocks for the characteristics.
5. Use the ECC to combine the blocks to produce abstract tests.
6. Choose values for each abstract test.
7. Add expected results.
8. Run the tests by hand.

In-class Exercise #2: ISP Tests

Design ISP tests for the simple web application [calculate](#).

Design the input domain model, abstract tests, and test values. Use the each choice criterion (ECC).

In your group, go through the following steps:

1. List the testable function or functions. **Add, subtract, multiply, divide, compute length**
2. List the inputs.
3. Define characteristics for the inputs.
4. Define blocks for the characteristics.
5. Use the ECC to combine the blocks to produce abstract tests.
6. Choose values for each abstract test.
7. Add expected results.
8. Run the tests by hand.

In-class Exercise #2: ISP Tests

Design ISP tests for the simple web application [calculate](#).

Design the input domain model, abstract tests, and test values. Use the each choice criterion (ECC).

In your group, go through the following steps:

1. List the testable function or functions. **Add, subtract, multiply, divide, compute length**
2. List the inputs. **Button click, first Val, second Val, name**
3. Define characteristics for the inputs.
4. Define blocks for the characteristics.
5. Use the ECC to combine the blocks to produce abstract tests.
6. Choose values for each abstract test.
7. Add expected results.
8. Run the tests by hand.

In-class Exercise #2: ISP Tests

Design ISP tests for the simple web application [calculate](#).

Design the input domain model, abstract tests, and test values. Use the each choice criterion (ECC).

In your group, go through the following steps:

1. List the testable function or functions. **Add, subtract, multiply, divide, compute length, reset**
2. List the inputs. **Button, first Val, second Val, name**
3. Define characteristics for the inputs.
C1: Button clicked
C2: Button value
C3: First val relation to 0
C4: Second val relation to 0
C5: Name length

In-class Exercise #2: ISP Tests

Define characteristics for the inputs.

C1: Button clicked

C2: Button value

C3: First val relation to zero

C4: Second val relation to zero

C5: Name length

Characteristic	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
Button clicked	True	False				
Button value	Add	Subtract	Multiply	Divide	Compute Length	Reset
First Val relation to zero	< 0	Equal to 0	> 0	No relation (empty)		
First Val relation to zero	< 0	Equal to 0	>0	No relation (empty)		
Name length	<=0	>= 1				

In-class Exercise #2: ISP Tests

Characteristic	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
Button clicked	True	False				
Button value	Add	Subtract	Multiply	Divide	Compute Length	Reset
First Val relation to zero	< 0	Equal to 0	> 0	No relation (empty)		
First Val relation to zero	< 0	Equal to 0	>0	No relation (empty)		
Name length	<=0	>= 1				

Use the ECC to combine the blocks to produce abstract tests.

(True, Add, <0, <0, <=0);

(False, Subtract, Equal to 0, Equal to 0, >=1) ;

(True, Multiply, >0, >0, <=0);

(False, Divide, No relation (empty), No relation (empty), >=1);

(True, Compute Length, >0, Equal to 0, <=0)

(False, Reset, >0, Equal to 0, >=1