

**2. Identify the variables that must be considered to test the method of interest.**

- Client.getAge()
- Client.isStudying()
- Client.isWorking()
- Client.livesWithParents()

**3. Identify the test values for each of the above variables using the three techniques seen in theory, specifying which one has been used for each one.**

Equivalence partitioning	Client.getAge() (int)	(-inf,0) [0,18) [18,25) [25,+inf)
	Client.isStudying() (boolean)	True False
	Client.isWorking() (boolean)	
	Client.livesWithParents() (boolean)	
Boundary values	Client.getAge() (int)	<b>Lightweight variant:</b> 0 18 25
		<b>Heavy variant:</b> Lightweight values -1 1 17 19 24 26
	Client.isStudying() (boolean)	<b>Lightweight and heavy variant:</b> True False
	Client.isWorking() (boolean)	
	Client.livesWithParents() (boolean)	
Error guessing	Client.getAge() (int)	-2000 5 20 30 2000
		null
	Client.isStudying() (boolean)	
	Client.isWorking() (boolean)	
	Client.livesWithParents() (boolean)	

**4. Calculate the maximum possible number of test cases generated from the test values (combinatorial).**

$14 \times 3^3 = 378$  test cases

**5. Define a set of test cases to meet "each use" (each value once).**

Client.getAge()	Client.isStudying()	Client.isWorking()	Client.livesWithParents()
0	True	False	null
18	False	null	True
25	null	True	False
-1	True	False	null
1	False	null	True
17	null	True	False
19	True	False	null
24	False	null	True
26	null	True	False
-2000	True	False	null
5	False	null	True
20	null	True	False
30	True	False	null
2000	False	null	True

**6. Define sets of tests to achieve pairwise coverage using the algorithm explained in class.**

Client.getAge() Client.isStudying() Client.isWorking() Client.livesWithParents()

17    False   null   True

24    True   False   True

0    null   False   null

1    null   null   False

18    null   True   True

1    False   True   null

-2000   True   null   null

25    False   False   False

5    True   True   null

26	True	null	False
30	True	True	False
2000	True	null	True
26	False	False	True
18	False	False	null
-2000	False	False	False
30	null	null	True
2000	False	False	False
17	True	True	False
1	True	False	True
19	False	null	True
-1	True	True	True
30	False	False	null
20	True	null	False
20	False	False	null
19	null	False	False
-2000	null	True	True
25	null	null	null
19	True	True	null
20	null	True	True
2000	null	True	null
17	null	False	null
-1	False	False	null
24	null	null	False
0	True	null	True
5	null	null	False

5	False	False	True
0	False	True	False
18	True	null	False
-1	null	null	False
25	True	True	True
24	False	True	null
26	null	True	null

**7. For code snippets that include decisions, propose a set of test cases to achieve decision coverage.**

**if** (client.getAge() < 18 && client.isStudying() && client.livesWithParents())

client.getAge() < 18	client.isStudying()	client.livesWithParents()	Decision	Test cases
T	T	T	T	[17, True, False, True]
T	T	F	F	[17, True, False, False]
T	F	T	F	
T	F	F	F	
F	T	T	F	
F	T	F	F	
F	F	T	F	
F	F	F	F	

**if** (client.getAge() < 25 && client.isStudying() && !client.livesWithParents())

client.getAge() < 25	client.isStudying()	!client.livesWithParents()	Decision	Test cases
T	T	T	T	[17, True, False, False]
T	T	F	F	[17, True, False, True]
T	F	T	F	
T	F	F	F	
F	T	T	F	
F	T	F	F	
F	F	T	F	
F	F	F	F	

**if** (client.getAge() >= 25 && client.isWorking())

client.getAge() >= 25	client.isWorking()	Decision	Test cases
T	T	T	[30, True, True, False]
T	F	F	
F	T	F	
F	F	F	[17, True, False, False]

**if** (client.livesWithParents())

client.livesWithParents()	Decision	Test cases
T	T	[30, True, True, True]
F	F	[30, True, True, False]

**if** (client.getAge() >= 18 && client.isWorking())

client.getAge() >= 18	client.isWorking()	Decision	Test cases
T	T	T	[19, True, True, True]
T	F	F	
F	T	F	
F	F	F	[17, True, False, False]

**if** (client.livesWithParents())

client.livesWithParents()	Decision	Test cases
T	T	[19, True, True, True]
F	F	[19, True, True, False]

**8. For code snippets that include decisions, propose a set of test cases to achieve MC/DC coverage.**

**if** (client.getAge() < 18 && client.isStudying() && client.livesWithParents())

client.getAge() < 18	client.isStudying()	client.livesWithParents()	Decision	Dominant Condition	Test cases
T	T	T	T	A,B,C	[17, True, False, True]
T	T	F	F	C	[17, True, False, False]
T	F	T	F	B	
T	F	F	F	B	[17, False, False, False]
F	T	T	F	A	[19, True, True, True]
F	T	F	F	A	
F	F	T	F	A	
F	F	F	F	A	

**if** (client.getAge() < 25 && client.isStudying() && !client.livesWithParents())

client.getAge() < 25	client.isStudying()	!client.livesWithParents()	Decision	Dominant Condition	Test cases
T	T	T	T	C	[19, True, False, False]
T	T	F	F	A, B	[17, True, False, True]
T	F	T	F	B	
T	F	F	F	B	[17, False, False, True]
F	T	T	F	A	[19, True, True, False]
F	T	F	F	A	
F	F	T	F	A	
F	F	F	F	A	

**if** (client.getAge() >= 25 && client.isWorking())

client.getAge() >= 25	client.isWorking()	Decision	Dominant Condition	Test cases
T	T	T	A, B	[30, True, True, False]
T	F	F		
F	T	F		
F	F	F	A, B	[17, True, False, False]

**if** (client.livesWithParents())

client.livesWithParents()	Decision	Test cases
T	T	[30, True, True, True]
F	F	[30, True, True, False]

**if** (client.getAge() >= 18 && client.isWorking())

client.getAge() >= 18	client.isWorking()	Decision	Dominant Condition	Test cases
T	T	T	A, B	[19, True, True, True]
T	F	F		
F	T	F		
F	F	F	A, B	[17, True, False, False]

**if** (client.livesWithParents())

client.livesWithParents()	Decision	Test cases
T	T	[19, True, True, True]
F	F	[19, True, True, False]

**9. Comment on the results of the number of test cases achieved in sections 4, 5, and 6. What could be said about the coverage achieved?**

As we got a maximum 378 of test cases in section 4, the coverage in:

- Section 5 →  $14/378 = 0.037 = 3.7\%$  of coverage
- Section 6 →  $42/378 = 0.111 = 11.1\%$  of coverage