Special Assignment

Software Testing April 29, 2020

Name: Mohammad Al-Tawil **ID:** 118256

Q.1 [CLO 3](2.5 marks) Suppose that income tax is calculated as follows: The first 20000 JD of annual income is tax-free. After that, the next 8000 is taxed 7%. After that, the next 6000 JD is taxed 14%. Any further income is taxed 20%. You are required to 1) Specify the equivalence partitions (valid and invalid), and 2) design test cases to cover all partitions.

Solution:

Invalid Partition	Valid (for free tax)	Valid (for 7%)	Valid (for 14%)	Valid (for 20%)
- $ 0.01	$ 0.00 $ 20.000	$ 20.001 $ 28.000	$ 28.001 $ 34.000	$ 34.001 ....

Valid Partition

Test Case 1

($0.00 - $20.000) for free

Test Case 2

($20.001 - $28.000) for 7%

Test Case 3

($28.001 - 34.000) for 14%

Test Case 4

($34.001 - more than) for 20%

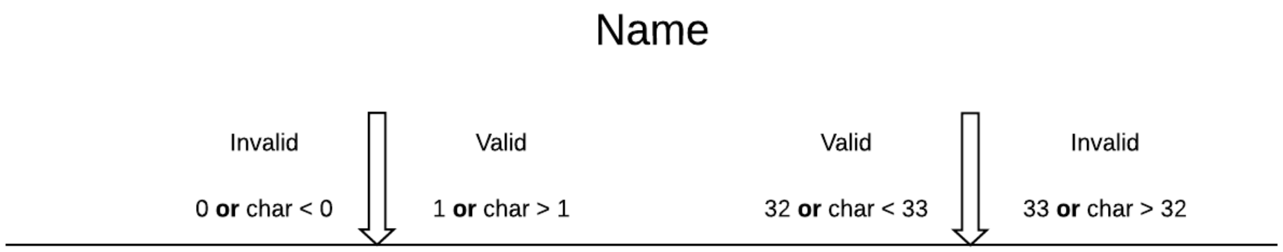
Invalid Partition

Annual Income < 0

Special Assignment

Q.2 [CLO 3](2.5 marks) A customer name field accepts strings with 1 – 32 alphabet characters. Design the condition template for the above software specification. Use the 2-Value approach for the boundary value analysis

Solution:



- **Valid Char: (A – Z , a – z)** and space if you want

Condition	Valid Partition	Invalid Partition	Valid Boundaries	Invalid Boundaries
name	1 to 32 char	<1	1	0 char
		>32	32	33 char

Special Assignment

Q.3 [CLO 3](4 marks) Given the following specification, provide a decision table and then provide a reduced decision table a credit rating program takes information about the customer such as age, employment type (Full time, part time Job), and education level (Degree, non-degree). The credit is considered:

- Very good if the customer (age>20, has full-time job, and has a degree);
- Good if the customer o (age>20, has full time, and non-degree) or o (age>20, has part-time job, and degree);
- Bad if the customer (age>20, has part-time, and non-degree).
- If the customer (age<=20) then no-credit

Solution:

Condition	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**	**Rule 5**	**Rule 6**	**Rule 7**	**Rule 8**
Age>20	**T**	**T**	**T**	**T**	**F**	**F**	**F**	**F**
Age<=20	**F**	**F**	**F**	**F**	**T**	**T**	**T**	**T**
Full time	**T**	**T**	**F**	**F**	**F**	**F**	**F**	**F**
Part time	**F**	**F**	**T**	**T**	**F**	**F**	**F**	**F**
Degree	**T**	**F**	**T**	**F**	**F**	**F**	**F**	**F**
Non-Degree	**F**	**T**	**F**	**T**	**F**	**F**	**F**	**F**
Action								
Credit	**T**	**T**	**T**	**T**	**F**	**F**	**F**	**F**
Very good	**T**	**F**	**F**	**F**	**F**	**F**	**F**	**F**
Good	**F**	**T**	**T**	**F**	**F**	**F**	**F**	**F**
Bad	**F**	**F**	**F**	**T**	**F**	**F**	**F**	**F**

Condition	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**	**Rule 5**
Age>20	**T**	**T**	**T**	**T**	**F**
Age<=20	**F**	**F**	**F**	**F**	**T**
Full time	**T**	**T**	**F**	**F**	**-**
Part time	**F**	**F**	**T**	**T**	**-**
Degree	**T**	**F**	**T**	**F**	**-**
Non-Degree	**F**	**T**	**F**	**T**	**-**
Action					
Credit	**T**	**T**	**T**	**T**	**F**
Very good	**T**	**F**	**F**	**F**	**-**
Good	**-**	**T**	**T**	**F**	**-**
Bad	**-**	**-**	**-**	**T**	**-**

Special Assignment

Software Testing April 29, 2020

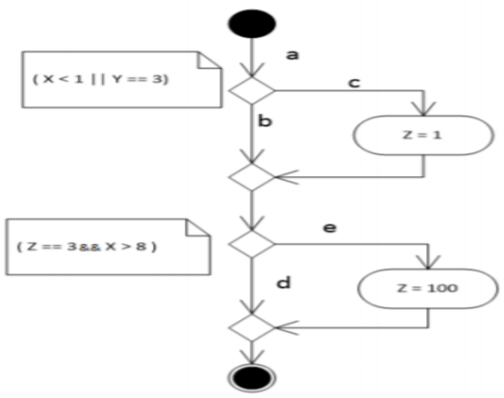
Q.4 [CLO 2](6 marks) Given the following code and its associated workflow, you are required to:
a) list all feasible paths
b) design test cases to cover each of the listed paths
c) write sufficient test cases so that all possible combinations of condition outcomes in each decision (multiple condition coverage)

```
void doSomething(int x, int y, int z){

if ( x < 1 || y == 3 ){
        z = 1;
}

if ( z == 3 && x > 8 ){
        z = 100;
}

}
```



Solution:

a) Feasible paths:

1- abd 2- abe 3- acd

b) cover each of the listed paths

Path	**Input**	**Output**
abd	x = 2 , y = 2 , z = 6	z = 6
abe	x = 10 , y = 4 , z = 3	z = 100
acd	x = 0 , y = 3 , z = 4	z = 1

c) Multiple condition coverage:

1- x<1 and y==3
2- x<1 and y!=3
3- x>=1 and y==3
4- x>=1 and y!=3
5- z==3 and x>8
6- z==3 and x<=8
7- z!=3 and x>8
8- z!=3 and x<=8

x = -1 , y = 3 , z = 2 Covers 1 , 8
x = -1 , y = 2 , z = 3 Covers 2 , 6
x = 10 , y = 2 , z = 2 Covers 4 , 7
x = 12 , y = 3 , z = 3 Covers 3 , 5