ECP

Guidelines for Equivalence Class Partitioning

- An input condition specifies a range [a, b]
  - one equivalence class for a $< X < b$, and
  - two other classes for $X < a$ and $X > b$ to test the system with invalid inputs
- An input condition specifies a set of values
  - one equivalence class for each element of the set $\{M_1\}$, $\{M_2\}$, ..., $\{M_N\}$, and
  - one equivalence class for elements outside the set $\{M_1,M_2, ...,M_N\}$
- Input condition specifies for each individual value
  - If the system handles each valid input differently then create one equivalence class for each valid input

Guidelines for Equivalence Class Partitioning

- An input condition specifies the number of valid values (Say N)
  - Create one equivalence class for **the correct number of inputs**
  - two equivalence classes for invalid inputs – **one for zero values and one for more than N values**
- An input condition specifies a “must be” value
  - Create one equivalence class for a “must be” value, and
  - one equivalence class for something that is not a “must be” value

Identification of Test Cases

Test cases for each equivalence class can be identified by:

- Assign a unique number to each EC

- For each EC with **valid input** that has not been covered by test cases yet, write a new test case covering **as many** uncovered EC as possible

- For each EC with **invalid input** that has not been covered by test cases, write a new test case that covers **one and only one** of the uncovered EC

Example-Adjusted Gross Income

Consider a software system that computes income tax based on adjusted gross income (AGI) according to the following rules:

- If AGI is between $1 and $29,500, the tax due is 22% of AGI.
- If AGI is between $29,501 and $58,500, the tax due is 27% of AGI.
- If AGI is between $58,501 and $100 billion, the tax due is 36% of AGI.

Example

<u>Condition 1</u>

$1 ≤ AGI ≤ $29,500, to derive two ECs:

EC1: $1 ≤ AGI ≤ $29,500; valid input.

EC2: AGI < 1; invalid input.

<u>Condition 2</u>

$29,501 ≤ AGI ≤ $58,500, to derive one EC:

EC3: $29,501 ≤ AGI ≤ $58,500; valid input.

<u>Condition 3</u>

$58,501 ≤ AGI ≤ $100 billion, to derive two ECs:

EC4: $58,501 ≤ AGI ≤ $100 billion; valid input.

EC5: AGI > $100 billion; invalid input.

Example

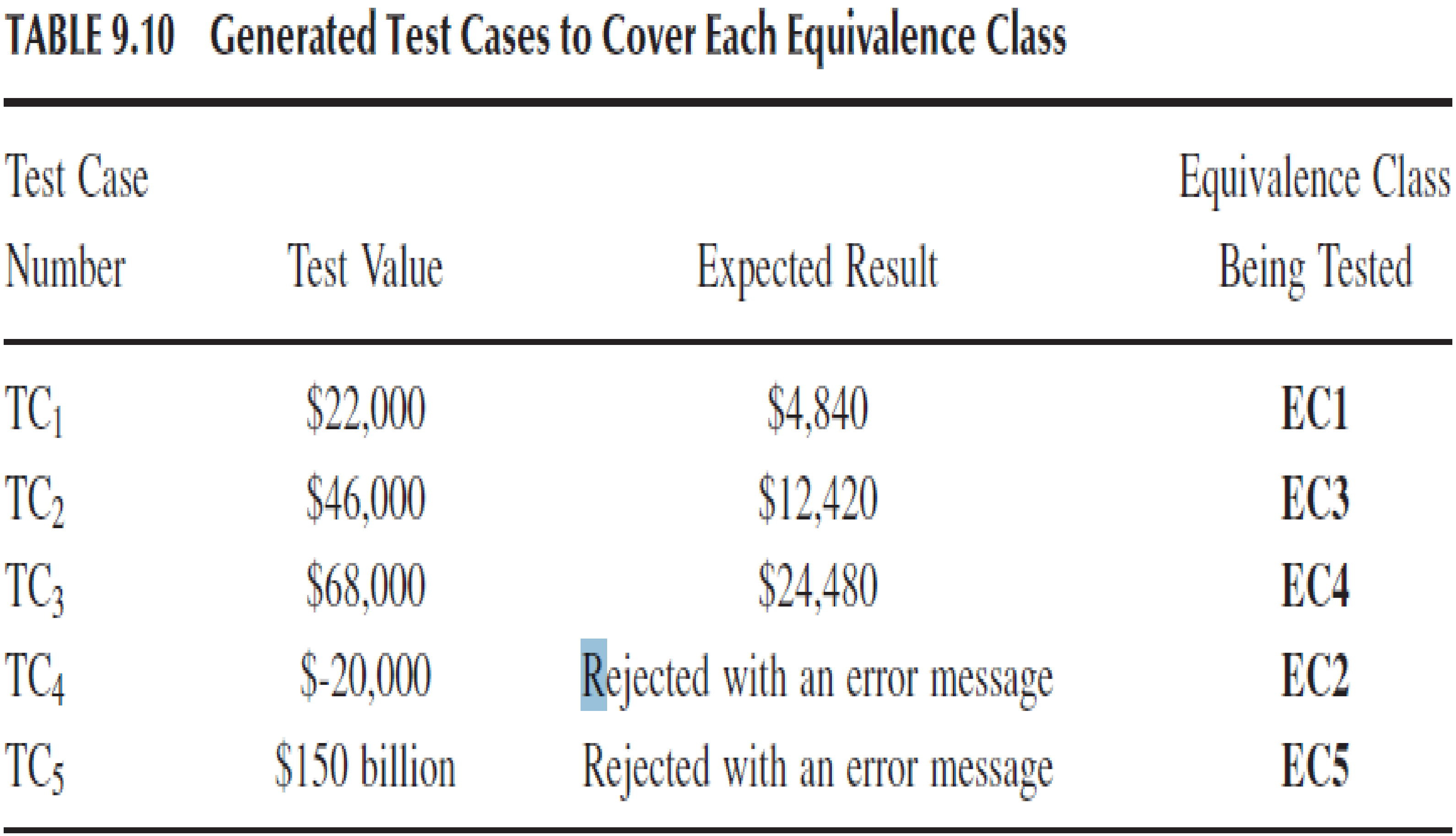

TABLE 9.10 Generated Test Cases to Cover Each Equivalence Class

Test Case			Equivalence Class
Number	Test Value	Expected Result	Being Tested
TC_1	$22,000	$4,840	EC1
TC_2	$46,000	$12,420	EC3
TC_3	$68,000	$24,480	EC4
TC_4	$-20,000	Rejected with an error message	EC2
TC_5	$150 billion	Rejected with an error message	EC5

Functional Specification 1: -

A login process allows user ID & password to authorize users.

From customer requirements user ID takes numerics in lower case from 4 to 16 characters long.

The password object takes alphabets in lower case from 4 to 8 characters long.

Prepare test case titles or scenario

Test Case Title 1: ***Verify user ID***

Boundary Value Analysis (Size)

Min-1 ----- 3 Characters -------Fail

Min ------- 4 Characters -------Pass

Min+1---- 5 Characters -------Pass

Max-1 ---15 Characters ------ Pass

Max -----16 Characters ------ Pass

Max+1- 17 Characters ------ Fail

Equivalence Class partition(Type)

Valid	*Invalid*
a - z	*A - Z*
0 – 9	*Special Chars.*
	Blank field.

Test case Title 2: ***Verify password***

Boundary Value Analysis (Size)

Min-1 ----- 3 Characters ---- Fail

Min ------- 4 Characters ---- Pass

Min+1 --- 5 Characters ---- Pass

Max-1---- 7 Characters ---- Pass

Max ----- 8 Characters ---- Pass

Max+1 – 9 Characters ---- Fail

Equivalence Class Partition (Type)

Valid	Invalid
a – z	*A - Z*
	0 - 9
	Special Chars
	Blank Field

Test Case Title 3: ***Verify Login Information***

User ID	**Password**	**Criteria**
Valid Value	*Valid Value*	*Pass*
Valid Value	*Invalid Value*	*Fail*
Invalid Value	*Valid Value*	*Fail*
Blank Value	*Valid Value*	*Fail*
Valid Value	*Blank Value*	*Fail*
Invalid Value	*Blank Value*	*Fail*
Blank Value	*Invalid Value*	*Fail*
Blank Value	*Blank Value*	*Fail*