

Homework 1 – Test Design - Daniela Fishwick

Task is:

Design tests applying test design techniques

Chosen case: Variant 1

1. Equivalence partitioning and Boundary value analysis

In a system designed to work out the tax to be paid:

- An employee has a \$1000 tax free salary.
- The next \$500 is taxed at 10%.
- The next \$4000 is taxed at 22%.
- Any further amount is taxed by 40%.
- Salary cannot be larger than \$6000.

1.1. *Build equivalence classes (partitions) based on given information*

The salary amounts were divided into the following equivalence classes (partitions) for testing the tax calculation system. The definition of them are as follows:

Class 0: Invalid salaries (zero or negative)

Definition: Salaries less than \$0.

Boundary: \$-1.

Example values: \$-1, \$-1000, \$-72.

Expected behavior: The system should handle this as an invalid input, possibly with an error message.

Class 1: Salaries below the tax-free threshold:

Definition: Salaries less than or equal to \$1000.

Boundaries: \$0, \$1000.

Example values: \$0, \$500, \$1000.

Expected behavior: No tax should be calculated.

Class 2: Salaries within the first tax bracket

Definition: Salaries greater than \$1000 and less than or equal to \$1500.

Boundaries: \$1001, \$1500.

Example values: \$1001, \$1250, \$1500.

Expected behavior: The portion of the salary above \$1000 should be taxed at 10%.

Class 3: Salaries within the second tax bracket

Definition: Salaries greater than \$1500 and less than or equal to \$5500.

Boundaries: \$1501, \$5500.

Example values: \$1501, \$3000, \$5500.

Expected behavior: The portion between \$1001 and \$1500 should be taxed at 10%, and the portion between \$1501 and \$5500 should be taxed at 22%.

Class 4: Salaries within the third tax bracket (maximum valid salary)

Definition: Salaries greater than \$5500 and less than or equal to \$6000.

Boundaries: \$5501, \$6000.

Example values: \$5501, \$5800, \$6000.

Expected behavior: The portion between \$1001 and \$1500 should be taxed at 10%, the portion between \$1501 and \$5500 should be taxed at 22%, and the portion above \$5500 (up to \$6000) should be taxed at 40%.

-

Class 5: Invalid salaries (above the allowed limit)

Definition: Salaries greater than \$6000.

Boundary: \$6001.

Example values: \$6001, \$7000, \$10000.

Expected behavior: The system should handle this as an invalid input, possibly with an error message.

1.2. Stand Out boundary values (BVA)

As was described in the answer above, the BVA are:

Class	BVA
0	\$-1
1	\$0 - \$1000
2	\$1001 - \$1500
3	\$1501 - \$5500
4	\$5501 - \$6000
5	\$6001

1.3. Cover requirements above by tests (write test cases' names and objectives) based on equivalence partitioning and boundary value analysis

The next table shows the test cases designed using equivalence partitioning (EP) and boundary value analysis (BVA) to cover the given salary tax requirements:

Test Case ID	Test Case Name	Objective
EP_01	Negative Salary Input	Verify the system handles a negative salary input correctly (Class 0: Invalid EP).
EP_02	Salary in Tax-Free Range	Verify no tax is calculated for a salary within the tax-free range (Class 1: Tax-Free EP).
EP_03	Salary in 10% Tax Bracket	Verify the system calculates tax at 10% for a salary in the 10% tax bracket (Class 2: 10% Taxed EP).
EP_04	Salary in 22% Tax Bracket	Verify the system calculates tax at 22% for a salary in the 22% tax bracket (Class 3: 22% Taxed EP).
EP_05	Salary in 40% Tax Bracket	Verify the system calculates tax at 40% for a salary in the 40% tax bracket (Class 4: 40% Taxed EP).
EP_06	Salary Over Limit Input	Verify the system handles a salary input over the limit correctly (Class 5: Invalid EP).
BVA_01	Boundary Value at Zero	Verify tax calculation for the lower boundary of the tax-free range -class 1- (\$0).
BVA_02	Boundary Value Below Tax-Free	Verify tax calculation for the value just below the tax-free threshold -class 1- (\$1).
BVA_03	Boundary Value At Tax-Free Limit	Verify tax calculation for the upper boundary of the tax-free range -class 1- (\$1000).
BVA_04	Boundary Value Above Tax-Free	Verify tax calculation for the value just above the tax-free threshold -class 2- (\$1001).
BVA_05	Boundary Value At 10% Limit	Verify tax calculation for the upper boundary of the 10% tax bracket -class 2- (\$1500).
BVA_06	Boundary Value Above 10%	Verify tax calculation for the value just above the 10% tax bracket -class 3- (\$1501).

BVA_07	Boundary Value At 22% Limit	Verify tax calculation for the upper boundary of the 22% tax bracket - <i>class 3-</i> (\$5500).
BVA_08	Boundary Value Above 22%	Verify tax calculation for the value just above the 22% tax bracket - <i>class 4-</i> (\$5501).
BVA_09	Boundary Value At 40% Limit	Verify tax calculation for the upper boundary of the 40% tax bracket - <i>class 4-</i> (\$6000).
BVA_10	Boundary Value Above 40% (Invalid)	Verify the system handles a value just above the salary limit - <i>class 5-</i> (\$6001) correctly (Invalid BVA).
BVA_11	Boundary Value Below Zero (Invalid)	Verify the system handles a negative boundary value - <i>class 0-</i> (e.g., -\$1) correctly (Invalid BVA).

2. Decision tables

You take a loan in a bank. The bank gives you loan application, where you can enter the amount of the monthly re-payment or the number of years you want to take to pay it back (the term of the loan). You should infill only one of the proposed fields. If you enter both, then you will get an error message.

2.1. Build decision table based on given information

First, the causes or input defined are:

- Monthly repayment entered
- Loan term

The outputs or effects described are:

- Loan application
- Error message

The resulting decision table is:

Decision Table: Loan Application Input Validation

Causes (inputs)	Rule 1	Rule 2	Rule 3	Rule 4
Repayment Amount Entered?	Y	Y	N	N
Loan Term (Years) Entered?	Y	N	Y	N
Effects (outputs)				
Loan Application Proceeds	N	Y	Y	N
Display Error Message	Y	N	N	N

Explanation of the Rules:

#	Conditions	Outcomes
1	Repayment Amount Entered? = Y AND Loan Term (Years) Entered? = Y	Loan Application Proceeds = N AND Display Error Message = Y
2	Repayment Amount Entered? = Y AND Loan Term (Years) Entered? = N	Loan Application Proceeds = Y AND Display Error Message = N
3	Repayment Amount Entered? = N AND Loan Term (Years) Entered? = Y	Loan Application Proceeds = Y AND Display Error Message = N
4	Repayment Amount Entered? = N AND Loan Term (Years) Entered? = N	Loan Application Proceeds = N.*

*Based on the explicit rule provided in the case, no error message is explicitly mentioned for this scenario in the initial requirements.

3. State transition

Customer chooses Arabica coffee from coffee machine. He selects specific sort of coffee (in this case Arabica), and enters money. If not enough money is entered, then machine will ask to enter more. If amount of money is ok, then machine will start doing coffee. If Arabica coffee is available, then customer will get coffee and his change in a minute. If there is no selected sort of coffee, then customer will get proper message and his money back.

3.1. Build state transition diagram based on given information

The state transition diagram for the coffee machine is built, focusing on the scenario where a customer chooses Arabica coffee.

The states identified are:

- **Waiting for Selection:** The initial state of the coffee machine, where it is ready for a customer to make a selection.
- **Arabica Selected:** The customer has chosen the "Arabica" coffee option.
- **Waiting for Money:** The machine is now waiting for the customer to insert money for the selected Arabica coffee.
- **Checking Money:** The machine is processing the entered amount of money.
- **Asking for More Money:** If the amount of money entered is insufficient for the Arabica coffee.
- **Making Coffee (Arabica):** The machine has received enough money and is now in the process of brewing the Arabica coffee.
- **Dispensing Coffee and Change:** The Arabica coffee is ready, and the machine is dispensing it along with any applicable change due to the customer.
- **Coffee Not Available/Returning Money:** The selected Arabica coffee is not available. The machine displays a message and returns the customer's money.

The events and conditions identified are:

- **Customer chooses Arabica coffee (event)**
- **Machine prompts for money (implicit event)**
- **Customer enters money (event)**
- **Not enough money is entered (condition)**
- **Customer enters more money (event)**
- **Amount of money is ok (condition)**
- **Arabica coffee is available (condition)**
- **Customer gets coffee and his change (event)**
- **There is no selected sort of coffee (Arabica) (condition)**
- **Customer will get proper message and his money back (event)**

Finally, the resulting state transition diagram is:

