School of Information Technology, Engineering, Mathematics and Physics

CS341: Software Quality Assurance and Testing

Semester 2, 2024

Assignment 1 – Software Testing

Course Coordinator: Vineet Singh

F2F

Group 9

Members:

Akshat Newal (S11214190)

Mohammed Kamal (S11171774)

Submission Date: 01/10/24

Question 1

Condition Table

Conditions	Valid Partition	Invalid Partition	Valid Boundary	Invalid Boundary

Loan Type	1-2	Any number other than 1,2	1-2	0,3
Loan Amount	$1< loan Amount (can be bigger than 500,000)	$0, non-numeric, negative values	$1, $99,999, $100,000, $499,999, $500,000	0, negative values, non-numeric
Loan Year	$1< Loan Year	$0, non-numeric, negative values	1, 5, 6, 10, 11,12	0, negative values, non-numeric

Task 1 – Equivalence Partitioning Table

Sl No	Test case name	Test Procedure	Condition (Loan Amount)	Loan Amount	Condition (Year Loan)	Year Loan	Expected Result (Interest Rate)	Result (Total Interest)
EPTC1	Home Loan	EP	LoanAmount < 100,000	50,000	YearLoan <=5	4	8%	(loanAmount * 0.08 * yearLoan)
EPTC2	Home Loan	EP	LoanAmount < 100,000	80,000	6 <= yearLoan <= 10	7	6.5%	(loanAmount * 0.065 * yearLoan)
EPTC3	Home Loan	EP	LoanAmount < 100,000	90,000	yearLoan >= 11	13	5.5%	(loanAmount * 0.055 * yearLoan)
EPTC4	Home Loan	EP	100,000 <= loanAmount < 500,000	150,000	yearLoan <= 10	8	6.5%	(loanAmount * 0.065 * yearLoan)
EPTC5	Home Loan	EP	loanAmount >= $500,000	600,000	yearLoan >= 11 years	12	5.5%	(loanAmount * 0.055 * yearLoan)
EPTC6	Home Loan	EP	LoanAmount < 100,000	0	YearLoan <=5	0	8%	0
EPTC7	Home Loan	EP	LoanAmount < 100,000	-15	YearLoan <=5	-4	8%	-1
EPTC8	Property Loan	EP	LoanAmount < 100,000	55,050	YearLoan <=5	3	12%	(loanAmount * 0.12 * yearLoan)
EPTC9	Property Loan	EP	LoanAmount < 100,000	85,005	6 <= yearLoan <= 10	8	8.5%	(loanAmount * 0.085 * yearLoan)
EPTC10	Property Loan	EP	LoanAmount < 100,000	97,060	yearLoan >= 11	12	7%	(loanAmount * 0.070 * yearLoan)
EPTC11	Property Loan	EP	100,000 <= loanAmount < 500,000	150,123	yearLoan < 10	9	8.5%	(loanAmount * 0.085 * yearLoan)
EPTC12	Property Loan	EP	loanAmount >= $500,000	621,000	yearLoan >= 11 years	15	7%	(loanAmount * 0.070 * yearLoan)
EPTC13	Property Loan	EP	LoanAmount < 100,000	0	YearLoan <=5	0	8%	0

EPTC14	Property Loan	EP	LoanAmount < 100,000	-15	YearLoan <=5	-4	8%	-1
EPTC15	Home Loan	EP	Any non-numeric letter or string characters	abc@/	Any non-numeric letter or string characters	abc @/	-	-1
EPTC16	Property Loan	EP	Any non-numeric letter or string characters	abc@/	Any non-numeric letter or string characters	abc @/	-	-1

The above mentioned table represents test cases using equivalence partitioning. The **EP** in EPTC refers to Equivalence partitioning and **TC** refers to test case. Based on the 5 loan conditions, we have conducted 14 test cases.

For Home Loans:

- Condition 1 (loanAmount < $100,000 and yearLoan <= 5 years – 8% interest rate) was covered by **EPTC1**
- Condition 2 (loanAmount < $100,000 and 6 years <= yearLoan <= 10 years – 6.5% interest rate) was covered by **EPTC2**
- Condition 3 (loanAmount < $100,000 and yearLoan >= 11 years – 5.5% interest rate) was covered by **EPTC3**
- Condition 4 ($100,000 <= loanAmount < $500,000 and yearLoan <= 10 years – 6.5% interest rate) was covered by **EPTC4**
- Condition 5 (loanAmount >= $500,000 and yearLoan >= 11 years – 5.5% interest rate) was covered by **EPTC5**
- Invalid boundaries were also tested in **EPTC6** and **EPTC7**, these referred to null numbers and negative integers
- Invalid boundary of non-numeric characters, “strings” were tested in **EPTC15**

For Property Loans:

- Condition 1 (loanAmount < $100,000 and yearLoan <= 5 years – 12% interest rate) was covered by **EPTC8**
- Condition 2 (loanAmount < $100,000 and 6 years <= yearLoan < 10 years – 8.5% interest rate) was covered by **EPTC9**
- Condition 3 (loanAmount < $100,000 and yearLoan >= 11 years – 7% interest rate) was covered by **EPTC10**
- Condition 4 $100,000 <= loanAmount < $500,000 and yearLoan < 10 years – 8.5% interest rate) was covered by **EPTC11**
- Condition 5 (loanAmount >= $500,000 and yearLoan >= 11 years – 7% interest rate) was covered by **EPTC12**

- Likewise, Invalid boundaries were also tested in **EPTC13** and **EPTC14**
- Invalid boundary of non-numeric characters, “strings” were tested in **EPTC16**

Question 1

Task 1 – Boundary Value Analysis Table

Sl No	Test case name	Test Procedure	Condition (Loan Amount)	Loan Amount	Condition (Year Loan)	Year Loan	Expected Result (Interest Rate)	Result (Total Interest)
BVATC1	Home Loan	BVA	LoanAmount < 100,000	1	YearLoan <=5	1	8%	(loanAmount * 0.08 * yearLoan)
BVATC2	Home Loan	BVA	LoanAmount < 100,000	99,999	YearLoan <=5	5	8%	(loanAmount * 0.08 * yearLoan)
BVATC3	Home Loan	BVA	LoanAmount < 100,000	1	6 <= yearLoan <= 10	6	6.5%	(loanAmount * 0.065 * yearLoan)
BVATC4	Home Loan	BVA	LoanAmount < 100,000	99,999	6 <= yearLoan <= 10	10	6.5%	(loanAmount * 0.065 * yearLoan)
BVATC5	Home Loan	BVA	LoanAmount < 100,000	1	yearLoan >= 11	11	5.5%	(loanAmount * 0.055 * yearLoan)
BVATC6	Home Loan	BVA	LoanAmount < 100,000	99,999	yearLoan >= 11	12	5.5%	(loanAmount * 0.055 * yearLoan)
BVATC7	Home Loan	BVA	$100,000 <= loanAmount < $500,000	100,000	YearLoan <=10	1	6.5%	(loanAmount * 0.065 * yearLoan)
BVATC8	Home Loan	BVA	$100,000 <= loanAmount < $500,000	499,999	YearLoan <=10	10	6.5%	(loanAmount * 0.065 * yearLoan)
BVATC9	Home Loan	BVA	loanAmount >= $500,000	500,000	YearLoan <=11	11	5.5%	(loanAmount * 0.065 * yearLoan)
BVATC10	Home Loan	BVA	loanAmount >= $500,000	500,001	YearLoan <=11	12	5.5%	(loanAmount * 0.065 * yearLoan)
BVATC11	Home Loan	BVA	LoanAmount < 100,000	0	Invalid	0	8%	0
BVATC12	Home Loan	BVA	LoanAmount < 100,000	-15	Invalid	-4	8%	-1
BVATC13	Property Loan	BVA	LoanAmount < 100,000	1	YearLoan <=5	1	12%	(loanAmount * 0.12 * yearLoan)
BVATC14	Property Loan	BVA	LoanAmount < 100,000	99,999	YearLoan <=5	5	12%	(loanAmount * 0.12 * yearLoan)

BVATC15	Property Loan	BVA	LoanAmount < 100,000	1	6 <= yearLoan <= 10	6	8.5%	(loanAmount * 0.085 * yearLoan)
BVATC16	Property Loan	BVA	LoanAmount < 100,000	99,999	6 <= yearLoan <= 10	10	8.5%	(loanAmount * 0.085 * yearLoan)
BVATC17	Property Loan	BVA	LoanAmount < 100,000	1	yearLoan >= 11	11	7%	(loanAmount * 0.07 * yearLoan)
BVATC18	Property Loan	BVA	LoanAmount < 100,000	99,999	yearLoan >= 11	12	7%	(loanAmount * 0.07 * yearLoan)
BVATC19	Property Loan	BVA	$100,000 <= loanAmount < $500,000	100,000	YearLoan < 10	1	8.5%	(loanAmount * 0.085 * yearLoan)
BVATC20	Property Loan	BVA	$100,000 <= loanAmount < $500,000	499,999	YearLoan < 10	9	8.5%	(loanAmount * 0.085 * yearLoan)
BVATC21	Property Loan	BVA	loanAmount >= $500,000	500,000	YearLoan >=11	11	7%	(loanAmount * 0.07 * yearLoan)
BVATC22	Property Loan	BVA	loanAmount >= $500,000	500,001	YearLoan >=11	12	7%	(loanAmount * 0.07 * yearLoan)
BVATC23	Home Loan	BVA	LoanAmount < 100,000	0	YearLoan <=5	0	8%	0
BVATC24	Home Loan	BVA	LoanAmount < 100,000	-15	YearLoan <=5	-4	8%	-1

The above mentioned table represents test cases using Boundary value analysis **(BVA)**. The **BVA** in BVATC refers to boundary value analysis and **TC** refers to test case. Based on the 5 loan conditions, we have conducted 24 test cases.

For Home Loans:

- Condition 1 (loanAmount < $100,000 and yearLoan <= 5 years – 8% interest rate) was covered by **BVATC1** and **BVATC2,** focusing on the lower and upper bounds
- Condition 2 (loanAmount < $100,000 and 6 years <= yearLoan <= 10 years – 6.5% interest rate) was covered by **BVATC3** and **BVATC4,** focusing on the lower and upper bounds
- Condition 3 (loanAmount < $100,000 and yearLoan >= 11 years – 5.5% interest rate) was covered by **BVATC5** and **BVATC6,** focusing on the lower and upper bounds
- Condition 4 ($100,000 <= loanAmount < $500,000 and yearLoan <= 10 years – 6.5% interest rate) was covered by **BVATC7** and **BVATC8,** focusing on the lower and upper bounds
- Condition 5 (loanAmount >= $500,000 and yearLoan >= 11 years – 5.5% interest rate) was covered by **BVATC9** and **BVATC10,** focusing on the lower and upper bounds

- Invalid boundaries were also tested in **BVATC11** and **BVATC12**, these referred to null numbers and negative integers

For Property Loans:

- Condition 1 (loanAmount < $100,000 and yearLoan <= 5 years – 12% interest rate) was covered by **BVATC13** and **BVATC14**
- Condition 2 (loanAmount < $100,000 and 6 years <= yearLoan < 10 years – 8.5% interest rate) was covered by **BVATC15** and **BVATC16**
- Condition 3 (loanAmount < $100,000 and yearLoan >= 11 years – 7% interest rate) was covered by **BVATC17** and **BVATC18**
- Condition 4 $100,000 <= loanAmount < $500,000 and yearLoan < 10 years – 8.5% interest rate) was covered by **BVATC19** and **BVATC20**
- Condition 5 (loanAmount >= $500,000 and yearLoan >= 11 years – 7% interest rate) was covered by **BVATC21** and **BVATC22**
- Likewise, Invalid boundaries were also tested in **BVATC23** and **BVATC24**

Question 1

Task 2 – Testing and Defect Tracking

1. Introduction

The purpose of this report is to document the unit tests developed for the `CalcInterest` class, which calculates loan interest based on the loan amount, loan term, and loan type. The tests are designed to ensure the correctness and robustness of the loan interest computation across different scenarios, including valid inputs, boundary values, and invalid inputs.

2. Test Objectives

The objectives of the unit tests are to:

- Verify that the `CalcInterest` program correctly computes loan interest for valid loan types, amounts, and terms.
- Validate the handling of boundary and ep cases such as loan amounts and terms at or near thresholds.
- Ensure that invalid input values are properly handled with appropriate error messages or exceptions.

3. Test Methodology

The testing methodology follows two key approaches:

1. **Boundary Value Analysis (BVA):** Testing inputs at the boundaries of valid ranges for loan amounts and terms.
2. **Equivalence Partitioning (EP):** Grouping inputs into valid and invalid partitions to reduce the number of test cases while maintaining high coverage.
3. **Condition Tests (CT):** This tests the conditions given by the client which has discrepancies, usually faulty or unsure conditions

4. Test Cases

4.1. Valid Input Cases

The following test cases cover scenarios where the input values are valid:

1. **Home Loans with Loan Amount < $100,000:**
   - Test cases with loan terms (years) below 5 years, between 6 and 10 years, and greater than 11 years.
2. **Home Loans at the Boundary of $100,000 and $500,000:**
   - Verifies the interest calculation when the loan amount is exactly at boundary values like $100,000 and $500,000.
3. **Property Loans:**
   - Tests for property loans with valid loan amounts and terms, including cases at the $500,000 boundary and for amounts exceeding $500,000.

4.2. Boundary and Conditional Test Cases

Boundary values and corner cases test the system's ability to handle inputs near the edges of valid input ranges:

- **Loan Amount at $100,000 and $500,000:** Ensures that the program handles transitions at critical boundaries correctly.
- **Loan Amount > $500,000:** Verifies that the system correctly computes interest for amounts larger than $500,000.

4.3. Invalid Input Cases

These test cases validate the program's ability to handle incorrect or invalid input gracefully:

- **Negative Loan Amounts and Terms:** Ensures that negative values for loan amounts or years are rejected, returning an error code `-1`.
- **Zero Loan Amounts and Terms:** Verifies that zero values are not allowed and handled appropriately.
- **Invalid Loan Types:** Tests for invalid loan types outside the expected range, ensuring the system returns an error.
- **Non-Numeric or Special Characters:** A test case was added to ensure that non-numeric input (e.g., "abc@/") results in a `NumberFormatException`.

5. Results

The unit tests passed successfully for both valid and invalid inputs. The program correctly handled boundary values and invalid input by returning error messages or raising exceptions where necessary. This ensures that the loan interest calculation is both robust and reliable.

6. Conclusion

The unit tests developed for the `CalcInterest` class provide comprehensive coverage of the functionality, ensuring correct behavior for various input conditions. The tests include valid cases, boundary values, corner cases, and invalid input handling, making the system more reliable and error-resistant. Future modifications to the code should be accompanied by corresponding updates to these tests to maintain test coverage and system integrity. This can be accessed at: https://github.com/abyss01701/CS341-Assignment-1

Unit Test Plan for BVA and EP

Sl No	Inputs			Expected Result		Actual Result		Result	Reference to Detailed Design / Spec Document
	Loan Type	Loan Amount	Loan Year	Interest	Total	Interest	Total		
EPTC1	1	50,000	4	$16,000	$66000	$16,000	$66,000	Pass	Question 1, Task 1
EPTC2	1	80,000	7	$36,400	$116,400.	$36,400	$116,400	Pass	Question 1, Task 1
EPTC3	1	90,000	13	$64,350	$154,350	$64,350	$154,350	Pass	Question 1, Task 1
EPTC4	1	150,000	8	$78,000	$228,000	$78,000	$228,000	Pass	Question 1, Task 1
EPTC5	1	600,000	12	$39,600	$99,600	396,000	$99,600	Pass	Question 1, Task 1
EPTC6	1	0	0	Message= ”Invalid Input”	Message= “Invalid Input”	$0	$0	Fail	Question 1, Task 1
EPTC7	1	-15	-4	Message: “Invalid. “	Message: “Invalid. “	$4.80	$-10.20	Fail	Question 1, Task 1
EPTC8	2	55,050	3	$19,818	$74,868	$19,818	$74868	Pass	Question 1, Task 1
EPTC9	2	85,005	8	$57,803.40	$142,808.40	$57,803.40	$142,808.40	Pass	Question 1, Task 1
EPTC10	2	97,060	12	$81,530.40	$178,590.40	$81,530.40	$178,590.40	Pass	Question 1, Task 1
EPTC11	2	150,123	9	$114,844.09	$264,967.10	$114,844.09	$264,967.10	Pass	Question 1, Task 1
EPTC12	2	621,000	15	$652,050.	$1,273,050	$652,050.	$1,273,050	Pass	Question 1, Task 1
EPTC13	2	0	0	Message= ”Invalid Input”	Message= “Invalid Input”	$0	$0	Fail	Question 1, Task 1
EPTC14	2	-17	-3	Message=” Invalid input..”	Message= “Invalid Input”	$4.08	$-12.92	Fail	Question 1, Task 1
EPTC15	1	abc@/	]	Message=” Invalid input”	Message=” Invalid input”	-1	-1	Fail	Question 1, Task 1
EPTC16	2	abc@/	]	Message=” Invalid input”	Message=” Invalid input”	-1	-1	Fail	Question 1, Task 1
BVATC1	1	1	1	$0.08	$1.08	$0.08	$1.08	Pass	Question 1, Task 1
BVATC2	1	99,999	5	$39,999.60	$139,998.60	$39,999.60	$139,998.60	Pass	Question 1, Task 1
BVATC3	1	1	6	$0.39	$1.39	$0.39	$1.39	Pass	Question 1, Task 1
BVATC4	1	99,999	10	$64,999.35	$164,998.35	$64,999.35	$164,998.35	Pass	Question 1, Task 1
BVATC5	1	1	11	$0.61	$1.61	$0.61	$1.61	Pass	Question 1, Task 1
BVATC6	1	99,999	12	$65,999.34	$165,998.34	$65,999.34	$165,998.34	Pass	Question 1, Task 1

BVATC7	1	100,000	1	$6500	$106,500.	$6500	$106,500	Pass	Question 1, Task 1
BVATC8	1	499,999	10	$324,999.35	$824,998.35	$324,999.35	$824,998.35	Pass	Question 1, Task 1
BVATC9	1	500,000	11	$302,500	$802,500.	$302,500	$802,500	Pass	Question 1, Task 1
BVATC10	1	500,001	12	$330,000.66	$830,001.66	$330,000.66	$830,001.66	Pass	Question 1, Task 1
BVATC11	1	0	0	Message=””Invalid input”	Message=””Invalid input”	$0	$0	Fail	Question 1, Task 1
BVATC12	1	-15	-4	Message=””Invalid input”	Message=””Invalid input”	$4.80	$-10.20	Fail	Question 1, Task 1
BVATC13	2	1	1	$1.12	$0.12	$0.12	$1.12	Pass	Question 1, Task 1
BVATC14	2	99,999	5	$59,999.40	$159,998.40	$59,999.40	$159,998.40	Pass	Question 1, Task 1
BVATC15	2	1	6	$0.51	$1.51	$0.51	$1.51	Pass	Question 1, Task 1
BVATC16	2	99,999	10	$69,999.30	$169,998.30	$69,999.30	$169,998.30	Pass	Question 1, Task 1
BVATC17	2	1	11	$0.77	$1.77	$0.77	$1.77	Pass	Question 1, Task 1
BVATC18	2	99,999	12	$83,999.16	$183,998.16	$83,999.16	$183,998.16	Pass	Question 1, Task 1
BVATC19	2	100,000	1	$8500	$108,500.	$8500	$108,500	Pass	Question 1, Task 1
BVATC20	2	499,999	9	$382,499.24	$882,498.24	$382,499.24	$882,498.24	Pass	Question 1, Task 1
BVATC21	2	500,000	11	$385,000	$885,000.	$385,000	$885,000	Pass	Question 1, Task 1
BVATC22	2	500,001	12	$420,000.84	$920,001.84	$420,000.84	$920,001.84	Pass	Question 1, Task 1
BVATC23	2	0	0	Message=””Invalid input”	Message=””Invalid input”	$0	$0	Fail	Question 1, Task 1
BVATC24	2	-15	-4	Message=””Invalid input”	Message=””Invalid input”	$4.80	$-10.20	Fail	Question 1, Task 1
CT1	1	750,000	**7**	Message=”Invalid loan condition”	Message=”Invalid loan condition”	Message=”Invalid loan condition”	Message=”Invalid loan condition”	Pass	
CT2	2	750,000	7	Message=”Invalid loan condition”	Message=”Invalid loan condition”	Message=”Invalid loan condition”	Message=”Invalid loan condition”	Pass	

Defect Tracking

Defect #	Description	Detected Stage	Affected Tests	Type of Defect	Severity	Impact	Priority	Injected Stage	Action Taken
1	Non-positive integer inputs for loan amount or loan years result in positive values instead of error	Testing	EPTC7, EPTC14, BVATC12, BVATC24	Logical	Moderate	Incorrect calculations	medium	Design	Added validation to check if loan amount or loan years are ≤ 0 and return -1 for invalid inputs.
2	Program crashes when user enters non-numeric characters for loan amount or loan years	Testing	EPTC15, EPTC16	Logical	High	System crash	High	Design	Added an if statement to handle string characters
3	Loan amounts over $500,000 with yearLoan < 11 return as invalid, though the condition is unclear.	Coding	CT1, CT2	Conditional	Moderate	Incorrect rejection	Medium	Requirements	Will confirm with the client about the conditions and report
4	Null inputs should return “Invalid input” message but instead returns 0	Testing	EPTC6, EPTC13, BVATC11, BVATC23		Low	Incorrect output	Low	Design	Added an if statement to handle null values
5	There is no upper limit for loan amount so user could request for an unlimited amount of loan	Coding	-	Conditional	Low	Users could ask for loans the organization doesn’t have the resources for	Low	Design	Will confirm with the client about the conditions and report

Question 2

Introduction:

This report outlines the test cases developed for the Simple Interest Calculator system, focusing on validating various teller operations (Calculate Loan Amount, Calculate Interest, Loan Balance) under different input conditions. The system consists of three fields: `loanType`, `loanAmount`, and `loanTerm`, and includes error handling for incorrect and missing inputs.

Overview of Test Cases:

The test cases are designed to cover **statement, condition, and logical coverage** by simulating both valid and invalid scenarios. The following operations were tested:

1. **Calculate Loan Amount**: This operation calculates the total loan amount based on user inputs. Test cases include scenarios where the loan amount is valid, below the minimum threshold, negative, or missing. Additionally, inputs such as non-numeric values and missing loan terms are tested to ensure proper validation and error handling.
2. **Calculate Interest**: Interest calculation is tested for boundary values, including valid loan amounts, amounts exactly at the boundary (e.g., $5,000), and amounts slightly above the boundary. The system's response to valid and invalid loan types and terms is also covered.
3. **Loan Balance**: This operation is tested to ensure fields are correctly disabled and the balance is retrieved for loans from the year 2020 or when certain conditions are met.

Key Test Cases:

- **Valid Inputs**: Test Case 1, where all fields are populated correctly, verifies the system calculates the loan amount successfully.
- **Boundary Values**: Test Cases 9 and 10 verify the system handles values at or just above the $5,000 boundary.
- **Invalid Inputs**: Test Cases 5, 6, 7, and 16 focus on handling negative values, missing inputs, and non-numeric entries.
- **Error Messages**: Test Cases 3, 6, and 15 ensure the system displays appropriate error messages such as "Unable to process," "Invalid Input," and "Input Required" under specific conditions.

Error Handling:

The system is tested for its ability to provide clear and user-friendly feedback when incorrect or incomplete data is submitted. This includes conditions where the loan amount is below the minimum allowed value, when input fields are left blank, or when non-numeric values are entered. Test Case 15 also checks for generic system errors that may arise during submission.

Logical Coverage Test Cases

	Operation	Inputs	Expected Output	Comments

Test case ID		Loan Type	Loan Amount	Loan Year		
1	Calculate Loan Amount	1	10000	5	Loan Amount field enabled; Loan calculated successfully	Statement coverage for valid input and enabled fields
2	Calculate Loan Amount	1	6000	5	Interest calculated and displayed	Statement coverage for interest calculation
3	Calculate Loan Amount	1	4000	5	Message: "Unable to process. Amount below 5,000."	Statement coverage for invalid loan amount (< 5000)
4	Loan Balance	1	disabled	disabled	Fields disabled; Balance retrieved for loan from 2020	Statement/condition coverage for Loan Balance selection
5	Calculate Loan Amount	1	-500	5	Message: "Invalid Input"	Logical coverage for invalid negative loan amount
6	Calculate Loan Amount	1	-	-	Message: "Input Required"	Condition coverage for missing input validation
7	Calculate Loan Amount	1	6000	-	Message: "Input Required"	Logical coverage for missing yearLoan input
8	Loan Balance	2	-	-	Fields disabled; Message: "Input Required"	Statement/condition coverage for missing inputs and Loan Balance
9	Calculate Interest	2	5000	5	Interest calculated and displayed	Condition coverage for valid input at exact boundary
10	Calculate Interest	2	5001	5	Loan Amount calculated successfully	Statement coverage for just above boundary (valid input)
11	Calculate Interest	2	6000	11	Interest calculated based on loan type and year	Condition path for interest calculation with valid inputs
12	Calculate Loan Amount	-	10000	5	Message: "Invalid Input"	Logical coverage for missing loan type
13	Loan Balance	2	6000	11	Fields disabled; Balance displayed if loan taken in 2020	Statement/condition coverage for Loan Balance retrieval
14	Loan Balance	1	6000	5	Fields disabled; Loan Balance retrieved	Logical coverage for balance retrieval with valid yearLoan
15		-	-	-	Message: "Something went wrong with the Submission"	Coverage for system error message when fields are left empty
16	Calculate Loan Amount	1	6000	abc	Message: "Invalid Input"	Logical coverage for non-numeric yearLoan
17	Calculate Loan Amount	1	abc	5	Message: "Invalid Input"	Logical coverage for non-numeric loanAmount

Question 3

Refer to Question 3.doc

Question 4

- Creating Test cases for a project based on the test techniques adopted.
- The application verifies if the user has legal rights to vote in Fiji.

Techniques:

The two techniques used to test this application are ***Boundary Value Analysis*** and ***Equivalence Partitioning.***

Boundary Value Analysis: BVA focuses on testing the values at the boundary of the input ranges. In this case, we will test the **age** inputs, where legal voters must be **21 years old** or older. BVA helps reveal defects at the boundaries of valid and invalid age inputs.

Equivalence Partitioning: EP divides inputs into equivalence classes that are expected to have the same behaviour. In this case, we will partition the inputs based on **legal status** and **country of residence**. This technique helps cover different combinations of valid and invalid input categories without needing to test every possible value.

Technique 1: Boundary Value Analysis (BVA)

Testing the age criteria (Birth Year | Birth Month) using Boundary Values

Test Case ID	Test Description	Input Values	Expected Results
TC1	Verify with exact legal age of 21 years.	Birth Year: Current Year - 21 Birth Month: Any	User should be able to vote.
TC2	Verify with user just turning 21 this month.	Birth Year: Current Year - 21 Birth Month: Current Month	User should be able to vote.
TC3	Verify with user who will turn 21 next month (boundary case).	Birth Year: Current Year - 21 Birth Month: Next Month	User should **not** be able to vote.
TC4	Verify with user turning 21 last month.	Birth Year: Current Year - 21 Birth Month: Previous Month	User should be able to vote.

TC5	Verify with user who is just below the legal age (20 years).	Birth Year: Current Year - 20 Birth Month: Any	User should **not** be able to vote.
TC6	Verify with user who is exactly 22 years old.	Birth Year: Current Year - 22 Birth Month: Any	User should be able to vote.
TC7	Verify with user who is exactly 19 years old.	Birth Year: Current Year - 19 Birth Month: Any	User should **not** be able to vote.
TC8	Verify with user aged exactly 50 years.	Birth Year: Current Year - 50 Birth Month: Any	User should be able to vote.
TC9	Verify with user aged 18 (minimum voting age boundary).	Birth Year: Current Year - 18 Birth Month: Any	User should **not** be able to vote.
TC10	Verify with invalid birth year (future date).	Birth Year: Current Year + 1 Birth Month: Any	Error message should appear, invalid input.

Technique 2: Equivalence Partitioning

Testing the voter eligibility based on the given status and country

Test Case ID	Test Description	Input Values	Expected Results
TC1	Verify with valid voter (Permanent Resident and in Fiji).	Status: Permanent Residence Country: Fiji	User should be able to vote.
TC2	Verify with valid voter (Citizen and in Fiji).	Status: Citizen Country: Fiji	User should be able to vote.
TC3	Verify with valid voter (Permanent Resident but outside Fiji).	Status: Permanent Residence Country: Not Fiji	User should be able to vote.
TC4	Verify with invalid voter (Student Visa in Fiji).	Status: Student Visa Country: Fiji	User should **not** be able to vote.
TC5	Verify with invalid voter (Work Permit in Fiji).	Status: Work Permit Country: Fiji	User should **not** be able to vote.

TC6	Verify with valid voter (Citizen but living outside Fiji).	Status: Citizen Country: Not Fiji	User should be able to vote.
TC7	Verify with invalid voter (Work Permit outside Fiji).	Status: Work Permit Country: Not Fiji	User should **not** be able to vote.
TC8	Verify with valid voter (Permanent Resident in another country).	Status: Permanent Residence Country: Not Fiji	User should be able to vote.
TC9	Verify with invalid voter (Student Visa and living outside Fiji).	Status: Student Visa Country: Not Fiji	User should **not** be able to vote.
TC10	Verify with invalid status and country (Work Permit outside Fiji).	Status: Work Permit Country: Not Fiji	User should **not** be able to vote.

Evaluation of Test Techniques:

1. **Boundary Value Analysis (BVA):**
   - **Coverage:** This technique is highly effective for the **age verification** part of the app, as it tests the critical boundary conditions where users are at the edge of the legal voting age (21 years).
   - **Why it's useful:** The BVA ensures that we capture edge cases that are likely to cause errors (e.g., users who are just below or above the legal age). It is more precise in testing age-related rules but does not cover all input combinations.
2. **Equivalence Partitioning (EP):**
   - **Coverage:** EP provides broad coverage for testing user **status** and **country** combinations, as it breaks down the possible inputs into valid and invalid categories and tests only one case from each partition.
   - **Why it's useful:** EP is efficient as it reduces the number of test cases while ensuring that every possible equivalence class (valid or invalid user statuses and countries) is tested. It provides better overall coverage for legal status and country validation compared to BVA.

To sum up, **Equivalence Partitioning (EP)** provides **greater coverage** in this scenario because it allows you to efficiently test different combinations of user statuses and countries that determine voter eligibility. It ensures that all valid and invalid groups are covered without excessive test cases, making it more comprehensive than BVA, which focuses primarily on boundary conditions related to age.