ASSIGNMENT 2

Name: Zeyam Hussain

Roll No: 24P-3111

Name: Muhammad Saad Ahmad

Roll No: 24P-3003

Course: Introduction to Software Engineering

Instructor: Sir Umer Haroon

Semester: Fall 2025

Project Overview:

In this project we worked on fixing and improving a legacy C++ loan/EMI calculator. The original code contained outdated structures, missing input validation, hardcoded values, and several risks such as integer overflow. The goal of the assignment was to modernize the project using Software Engineering practices. We performed the following major task:

1) Fixed functional and logical bugs
2) We have added input validation
3) Replace double with long double to avoid overflow
4) Removed hardcoded values and added config.txt
5) Refactored the code into separate modules (Loan.h, Loan.cpp, main.cpp)
6) Added GoogleTest unit tests
7) Added documentation using Doxygen
8) Managed the project using GitHub and a structured branch workflow

CODE SNIPPETS OF BUG FIXES:

Input Validation:

```
while (loan_amount <= 0) {
    cout << "Invalid. Enter positive value: ";
    cin >> loan_amount;
```

The original code allowed negative and zero loaned amounts causing invalid EMI calculation. Now if the loan is zero or negative it will tell the user to enter a positive value.

Configuration File System:

```
100000
5.5
20
```

```
void readConfig(long double &loan, long double &rate, long double &years) {
    ifstream config("../config.txt");
    readConfigFromStream(config, loan, rate, years);
}
```

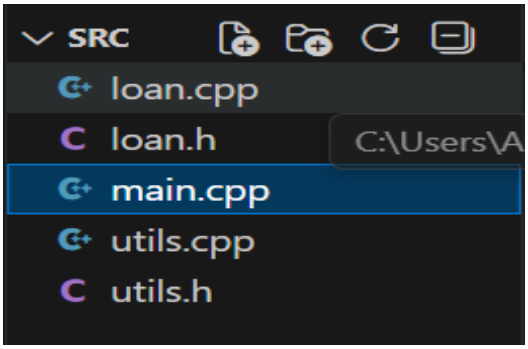
The calculator originally had hardcoded default values which we eliminated by refactoring them this makes the application more flexible.

Overflow fixing:

```
long double default_loan =
long double default_rate =
long double default_years =
```

Double was causing overflow due to which we changed the major variables from double to long double.

Refactoring into multiple files:



We have split the project into different files to improve the readability and maintainability so our code can be organized.

Doxygen Documentation:

```
/**
 * @brief Calculates total amount (principal + interest)
 * @return Total repayment amount
 */
long double Loan::getTotalAmount() const {
    long double annual_rate = interest_rate / 100.0L;
    long double total_interest_factor = years * annual_rate;
    return loan_amount + (loan_amount * total_interest_factor);
}
```

We have added documentation using Doxygen format which makes our codes understandable for others.

Test case condition:

What the Tests Cover:

Test 1: Normal EMI Calculation

Tests standard loan scenario

Verifies total amount, interest, and monthly payment calculations

Uses realistic values: $100,000 at 5% for 20 years

Test 2: Zero Interest Rate

Tests edge case with 0% interest

Ensures calculations handle zero interest correctly

Total amount should equal principal

Test 3: Large Tenure Without Overflow

Tests 100-year loan to check for overflow issues

Verifies all results are finite numbers

Confirms calculations work with extreme values

Test 4: Very Small Loan Amount

Tests boundary case with small principal

Ensures calculations work with minimal amounts

Test 5: Config File Reading

Tests configuration reading functionality

Verifies values are valid (non-negative)

Test 6: Edge Case Calculations

Tests unusual scenarios like 1-month loans

Tests high interest rates

Ensures no crashes or infinite values

Key Testing Concepts Explained:

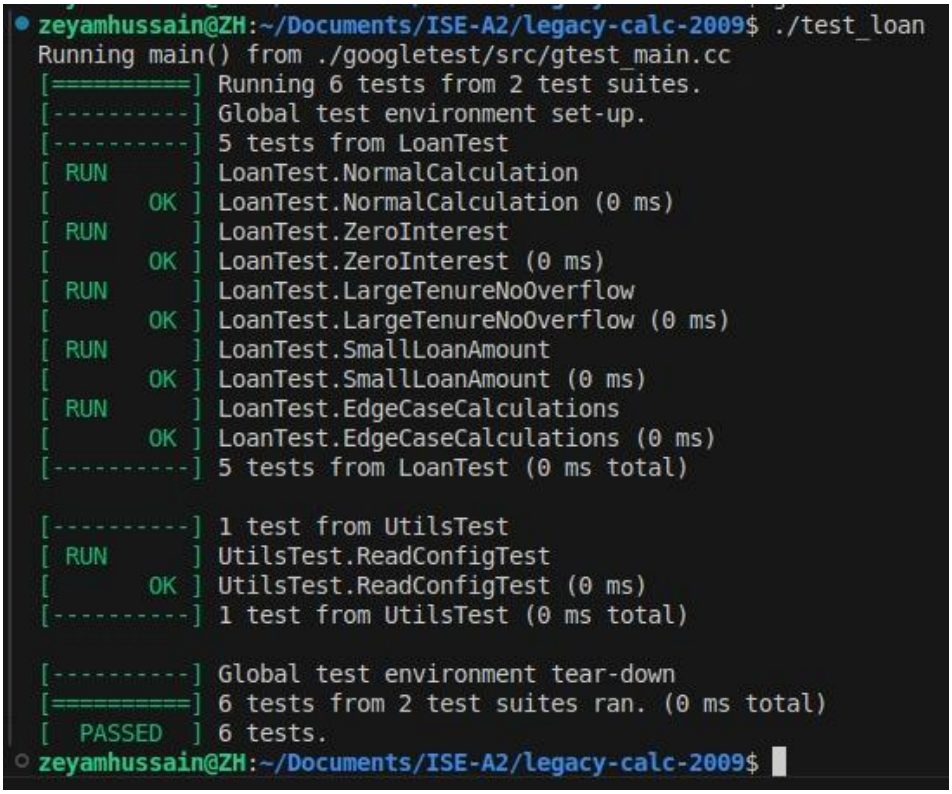
EXPECT_NEAR: Used for floating-point comparisons with tolerance

EXPECT_TRUE/GT: Used for boolean conditions and comparisons

std::isfinite: Checks that calculations don't produce overflow/underflow

Test Fixtures: Each TEST() macro creates a separate test case

Edge Cases: Testing boundaries and unusual inputs


zeyamhussain@ZH:~/Documents/ISE-A2/legacy-calc-2009$ ./test_loan
Running main() from ./googletest/src/gtest_main.cc
[==========] Running 6 tests from 2 test suites.
[----------] Global test environment set-up.
[----------] 5 tests from LoanTest
[RUN   ] LoanTest.NormalCalculation
[OK] LoanTest.NormalCalculation (0 ms)
[RUN   ] LoanTest.ZeroInterest
[OK] LoanTest.ZeroInterest (0 ms)
[RUN   ] LoanTest.LargeTenureNoOverflow
[OK] LoanTest.LargeTenureNoOverflow (0 ms)
[RUN   ] LoanTest.SmallLoanAmount
[OK] LoanTest.SmallLoanAmount (0 ms)
[RUN   ] LoanTest.EdgeCaseCalculations
[OK] LoanTest.EdgeCaseCalculations (0 ms)
[----------] 5 tests from LoanTest (0 ms total)
[----------] 1 test from UtilsTest
[RUN   ] UtilsTest.ReadConfigTest
[OK] UtilsTest.ReadConfigTest (0 ms)
[----------] 1 test from UtilsTest (0 ms total)
[----------] Global test environment tear-down
[==========] 6 tests from 2 test suites ran. (0 ms total)
[PASSED] 6 tests.
zeyamhussain@ZH:~/Documents/ISE-A2/legacy-calc-2009$

Git commit history:

```
$ git log
commit 1c6cdced50f428838c7adcfa9f318621f732f049 (HEAD -> main, origin/main, origin/HEAD)
Author: zeyam <zeyamhussain24@gmail.com>
Date:   Tue Nov 25 16:22:30 2025 +0000

    Successfully format the files and remove the errors.

commit d9d0ec1e23297fdb53cd66183a0858e46e22fbcb
Author: zeyam <zeyamhussain24@gmail.com>
Date:   Tue Nov 25 06:59:55 2025 +0000

    Successfully Add Doxygen documentation.

commit 6ec55b5d190e8def874429e75921446cf96e438e
Author: zeyam <zeyamhussain24@gmail.com>
Date:   Tue Nov 25 06:37:29 2025 +0000

    Successfully done the Unit testing.

commit 2a7e165e821481ae465cc2b184a2390caa9951f1
Author: zeyam <zeyamhussain24@gmail.com>
Date:   Tue Nov 25 06:09:57 2025 +0000

    Successfully Remove all the error in the code.

commit c539be9dff56b8db63eafb99eb799f512958616a
Author: Saad <muhammadsaadahmad01@gmail.com>
Date:   Tue Nov 25 00:12:23 2025 +0500

    Clean up output formatting using consistent

commit 0969728ff355deb008e5583440cec084cf9e78d9
Author: Saad <muhammadsaadahmad01@gmail.com>
Date:   Mon Nov 24 23:58:37 2025 +0500

    code has been seperated into multiple files

commit 02a3fa9af7af537df19e2c773fc92ccd1d71e92f
Author: zeyam <zeyamhussain24@gmail.com>
Date:   Mon Nov 24 18:42:16 2025 +0000

    Successfully remove the hard coded value from the code

commit 8e3e06fc1515939000efcbad3d3c14cda53399701
Author: zeyam <zeyamhussain24@gmail.com>
Date:   Mon Nov 24 18:33:13 2025 +0000

    Successfully handle the overflow problem

commit 869aad30bd6fde3c4ec0ee1344e306d7971f21ba
Author: zeyam <zeyamhussain24@gmail.com>
Date:   Mon Nov 24 18:21:56 2025 +0000
```