### ***EMI Calculator Workflow***

This EMI Calculator program helps users calculate the Equated Monthly Installment (EMI) for a loan based on the loan amount (principal), annual interest rate, and loan duration (in years). This program also validates the input for correctness and handles exceptions such as invalid interest rates or loan durations. It writes results to an output file and logs the entire process.

The program is divided into four components:

1. EMI Calculation Logic (emi\_calculator.py)
2. Custom Exception Classes (exceptions.py)
3. Decorators (decorators.py)
4. Main Program Logic (main.py)
5. Test Case Runner (test\_cases.py)

### ***EMI Calculation Logic (emi\_calculator.py)***

The function calculate\_monthly\_installment calculates the EMI using the following formula:

EMI=P⋅r⋅((1+r)^n) / ((1+r)^n)-1

Where:

1. P is the principal loan amount.
2. r is the monthly interest rate, which is the annual rate divided by 12 and then converted to a decimal by dividing by 100.
3. n is the total number of payments (loan duration in years multiplied by 12).

The function ensures:

1. If the interest rate is 0, the EMI is simply the principal divided by the total number of payments (monthly installments).
2. Returns the calculated EMI value.

### ***Custom Exception Classes (exceptions.py)***

There are two custom exception classes:

1. InvalidRateException: Raised when the annual interest rate is invalid (negative values).
2. InvalidLoanDurationException: Raised when the loan duration is invalid (less than or equal to 0, or greater than the maximum allowed value of 50 years).

These exceptions help in validating user input and maintaining the integrity of the EMI calculation process.

***Decorators (decorators.py)***

The retry\_on\_failure decorator is used to make the code more modular and easy to maintain.

### ***Main Program Logic (main.py)***

The main program:

1. Prompts the user to input the loan amount, interest rate, and loan duration.
2. Validates the input for errors like negative interest rates or loan durations that are out of the allowed range (1 to 50 years).
3. Calculates the EMI using the calculate\_monthly\_installment function.
4. Writes the results to an output file (LoanCalculatorOutput.txt) and logs them.
5. If an error occurs, the program logs the error and writes it to the file. The user has up to three attempts to enter valid data.
6. If all attempts are exhausted, the program exits.

***The workflow:***

1. Input is taken from the user (principal, interest rate, and loan duration).
2. Validations are performed. If validation fails, an exception is raised, and the user is prompted to try again.
3. Upon successful input, the EMI is calculated and logged to a file.
4. The program continues until the user enters valid data or exhausts their attempts.

### ***Test Case Runner (test\_cases.py)***

This module runs a set of predefined test cases for EMI calculation and writes the results to a file (TestCaseOutput.txt).

***The workflow:***

1. The program asks the user how many test cases they wish to run.
2. For each test case, the user is prompted to input the principal amount, interest rate, and loan duration.
3. The program validates the inputs:
   1. The interest rate must be positive.
   2. The loan duration must be between 1 and 30 years.
4. If valid, the EMI is calculated and written to the file.
5. If the validation fails, an error message is logged and written to the file.

### ***Execution Flow***

1. Start the Program:
   1. When the program is executed, main.py starts the process by calling the main() function.
   2. The user is prompted to input the loan details.
2. Input Validation:
   1. The program ensures that the principal, annual interest rate, and loan duration are within valid ranges.
   2. Invalid inputs trigger custom exceptions or assertion errors, which are logged and written to the output file.
3. EMI Calculation:
   1. If the inputs are valid, the EMI is calculated using the formula and logged.
   2. The result is written to the output file (LoanCalculatorOutput.txt).
4. Test Cases Execution:
   1. The program may also run predefined test cases to validate various input scenarios.
   2. The results of each test case are logged and saved to TestCaseOutput.txt.
5. File Output:
   1. All results and errors are logged both to a file and the console for tracking.
   2. The results include both the EMI calculations and any validation or input errors.

### ***Error Handling and Logging***

1. Errors are captured using custom exceptions (InvalidRateException, InvalidLoanDurationException) and assertions.
2. The program logs every step: from the start of the program to successful EMI calculations or errors.
3. If all user attempts are exhausted, the program logs an error and exits gracefully.
4. Every test case run logs the outcome or validation failure to the output file.

### ***Conclusion***

The EMI Calculator is a robust program that helps users calculate their monthly loan installments. It ensures the validity of inputs, calculates the EMI using a formula, handles errors gracefully, and logs the results for transparency. The test case runner provides a way to validate the program under different input scenarios, ensuring reliability and correctness.