

Project - “Money Lender”

CSE-310

(Term January-May 2023)



Team Members:

Amaan Ahmad (12107450)

Roll.No(RK21PKB61)

Harsh Mahawar (12105008)

Roll.No(RK21PKA30)

Neelabh Nagle (12108767)

Roll.No(RK21PKB50)

Section: K21PK

Subject: Programming In Java

School of Computer Science & Engineering

Lovely Professional University, Jalandhar

Introduction:

The Money Lender Calculator is a command-line tool developed in Java that allows users to calculate and visualize the details of a loan. The tool takes in user inputs such as the loan amount, interest rate, loan term, payment frequency, and any additional or lump sum payments, and outputs a detailed amortization schedule that shows the payment schedule, interest paid, principal paid, and remaining balance. The tool also calculates the total payment, total interest paid, and the total interest saved by making additional payments. The Money Lender Calculator is designed with an object-oriented approach to encapsulate the loan details and provide a more organized and modular code structure. With its intuitive and user-friendly interface, the Money Lender Calculator is a useful tool for anyone looking to plan and visualize the details of a loan.

The Basic Functionalities:

The Money Lender Calculator has the following main functionalities:

Loan Details Input: The tool takes user inputs for the loan amount, interest rate, loan term, payment frequency (monthly, weekly, or bi-weekly), and any additional or lump sum payments.

Amortization Schedule: The tool calculates and displays a detailed amortization schedule that shows the payment schedule, interest paid, principal paid, and remaining balance for each payment period.

Total Payment and Interest Calculation: The tool calculates and displays the total payment and total interest paid over the loan term.

Interest Saved Calculation: The tool calculates and displays the total interest saved by making additional payments.

Additional Payment: The tool allows for additional monthly payments to be made, which can be used to calculate the total interest saved.

Payment Frequency: The tool allows for payments to be made on a weekly or bi-weekly basis, in addition to the default monthly payment frequency.

Object-Oriented Design: The tool is designed with an object-oriented approach to encapsulate the loan details and provide a more organized and modular code structure.

Overall, the Money Lender Calculator is a comprehensive tool for loan planning and visualization that provides users with a detailed understanding of their loan repayment schedule, total payment, and interest paid.

Role And Responsibility of Each Member:

Amaan Ahmad :61

Develop a user-friendly command-line interface for the Money Lender Calculator.
Create a menu-based system that guides the user through the loan details input process.
Display the amortization schedule, total payment, total interest, and interest saved in an easy-to-read format.
Allow for user inputs for payment frequency and additional monthly payments.

Harsh Mahawar :30

Develop the core loan calculation logic using Java.
Create classes to encapsulate the loan details and payment schedule.
Implement the total payment and total interest calculation.
Implement the interest saved calculation for additional monthly payments.

Neelabh Nagle:50

Develop test cases to ensure the accuracy of the loan calculations and payment schedule.
Conduct functional testing of the Money Lender Calculator to ensure it performs as expected.
Identify and report any issues or bugs found during testing.
Work closely with the frontend and backend developers to ensure a high-quality product is delivered to the end user.

ScreenShots:

```
Welcome to the Money Lender Calculator!
Please enter the loan amount: 100000
Please enter the interest rate (%): 23
Please enter the loan term (years): 3
Please enter the payment frequency (m, w, or b): m
Please enter any lump sum payment (if applicable): 0
Please enter any additional monthly payment (if applicable): 0

Loan amount: $100000.00
Payment frequency: %s
Loan term: 3.00 years
Interest rate: 23.00%
Total interest paid: $0.00

Amortization Schedule:
-----
Payment#    Payment    Interest    Principal    Balance
1           $3870.97    $1916.67    $1954.31    $98045.69
2           $3870.97    $1879.21    $1991.76    $96053.93
3           $3870.97    $1841.03    $2029.94    $94023.99
4           $3870.97    $1802.13    $2068.85    $91955.15
5           $3870.97    $1762.47    $2108.50    $89846.65
6           $3870.97    $1722.06    $2148.91    $87697.74
```

```
Welcome to the Money Lender Calculator!
Please enter the loan amount: 200000
Please enter the interest rate (%): 25
Please enter the loan term (years): 3
Please enter the payment frequency (m, w, or b): w
Please enter any lump sum payment (if applicable): 0
Please enter any additional monthly payment (if applicable): 0

Loan amount: $200000.00
Payment frequency: %s
Loan term: 3.00 years
Interest rate: 25.00%
Total interest paid: $0.00

Amortization Schedule:
-----
Payment#    Payment    Interest    Principal    Balance
1           $70882.47    $4166.67    $66715.80    $133284.20
2           $70882.47    $2776.75    $68105.71    $65178.49
3           $66536.38    $1357.89    $65178.49    $0.00

Total payment: $208301.31
```

```

18           $3870.97    $1172.23    $2698.74    $58461.20
19           $3870.97    $1120.51    $2750.47    $55710.74
20           $3870.97    $1067.79    $2803.18    $52907.55
21           $3870.97    $1014.06    $2856.91    $50050.64
22           $3870.97    $959.30     $2911.67    $47138.97
23           $3870.97    $903.50     $2967.48    $44171.50
24           $3870.97    $846.62     $3024.35    $41147.15
25           $3870.97    $788.65     $3082.32    $38064.83
26           $3870.97    $729.58     $3141.40    $34923.43
27           $3870.97    $669.37     $3201.61    $31721.83
28           $3870.97    $608.00     $3262.97    $28458.85
29           $3870.97    $545.46     $3325.51    $25133.34
30           $3870.97    $481.72     $3389.25    $21744.09
31           $3870.97    $416.76     $3454.21    $18289.88
32           $3870.97    $350.56     $3520.42    $14769.47
33           $3870.97    $283.08     $3587.89    $11181.58
34           $3870.97    $214.31     $3656.66    $7524.92
35           $3870.97    $144.23     $3726.74    $3798.17
36           $3870.97    $72.80      $3798.17    $0.00

Total payment: $139355.00
Total interest paid: $39355.00
PS C:\Users\vashu\OneDrive\Desktop\javapeojects\money_lender>
```

Code :

```
import java.util.Scanner;

public class MoneyLender {
    private double loanAmount;
    private double interestRate;
    private double loanTerm;
    private String paymentFrequency;
    private double lumpSumPayment;
    private double additionalMonthlyPayment;

    public void setLoanAmount(double loanAmount) {
        this.loanAmount = loanAmount;
    }

    public void setInterestRate(double interestRate) {
        this.interestRate = interestRate;
    }

    public void setLoanTerm(double loanTerm) {
        this.loanTerm = loanTerm;
    }

    public void setPaymentFrequency(String paymentFrequency) {
        this.paymentFrequency = paymentFrequency;
    }

    public void setLumpSumPayment(double lumpSumPayment) {
        this.lumpSumPayment = lumpSumPayment;
    }

    public void setAdditionalMonthlyPayment(double
additionalMonthlyPayment) {
        this.additionalMonthlyPayment = additionalMonthlyPayment;
    }

    public double getLoanAmount() {
        return loanAmount;
    }

    public double getInterestRate() {
```

```

        return interestRate;
    }

    public double getLoanTerm() {
        return loanTerm;
    }

    public String getPaymentFrequency() {
        return paymentFrequency;
    }

    public double getLumpSumPayment() {
        return lumpSumPayment;
    }

    public double getAdditionalMonthlyPayment() {
        return additionalMonthlyPayment;
    }

    public void calculateLoan() {
        double moneyAvailable = loanAmount - lumpSumPayment;

        if (moneyAvailable <= 0) {
            System.out.println("You do not need a loan.");
            return;
        }

        double monthlyInterestRate = interestRate / 100 / 12;

        double monthlyPayment;
        if (paymentFrequency.equals("m")) {
            monthlyPayment = moneyAvailable * (monthlyInterestRate /
(1 - Math.pow(1 + monthlyInterestRate, -loanTerm * 12)));
        } else if (paymentFrequency.equals("w")) {
            monthlyPayment = moneyAvailable * (monthlyInterestRate /
(1 - Math.pow(1 + monthlyInterestRate, -loanTerm * 52 / 12)));
            monthlyPayment *= 4;
        } else {
            monthlyPayment = moneyAvailable * (monthlyInterestRate /
(1 - Math.pow(1 + monthlyInterestRate, -loanTerm * 26 / 12)));
            monthlyPayment *= 2;
        }
    }

```

```

        if (lumpSumPayment > 0) {
            System.out.printf("Lump sum payment of $%.2f
applied.%n", lumpSumPayment);
        }

        if (additionalMonthlyPayment > 0) {
            System.out.printf("Additional monthly payment of $%.2f
applied.%n", additionalMonthlyPayment);
        }

        double newMonthlyPayment;
        if (paymentFrequency.equals("m")) {
            newMonthlyPayment = (loanAmount - lumpSumPayment) /
loanTerm + additionalMonthlyPayment;
        } else if (paymentFrequency.equals("w")) {
            newMonthlyPayment = (loanAmount - lumpSumPayment) * (1 +
interestRate / 100 / 52);
            newMonthlyPayment = newMonthlyPayment / 4 +
additionalMonthlyPayment;
        } else {
            newMonthlyPayment = (loanAmount - lumpSumPayment) * (1 +
interestRate / 100 / 26);
            newMonthlyPayment = newMonthlyPayment / 2 +
additionalMonthlyPayment;
        }

        double newTotalInterest = loanAmount - moneyAvailable;

        System.out.printf("Loan amount: $%.2f%n", loanAmount);
        System.out.printf("Payment frequency: %s%n",
paymentFrequency.equals("m") ? "Monthly" :
paymentFrequency.equals("w") ? "Weekly" : "Bi-weekly");
        System.out.printf("Loan term: %.2f years%n", loanTerm);
        System.out.printf("Interest rate: %.2f%%n", interestRate);
        System.out.printf("Total interest paid: $%.2f%n",
newTotalInterest);
        double totalPayment = 0;
        double totalInterestPaid = 0;
        double remainingBalance = loanAmount;

        System.out.println("\nAmortization Schedule:");

```

```

        System.out.println("-----");
        System.out.println("Payment#\tPayment\t\tInterest\tPrincipal
\tBalance");

        for (int i = 1; i <= loanTerm * 12; i++) {
            double monthlyInterest = remainingBalance *
monthlyInterestRate;
            double monthlyPrincipal = monthlyPayment -
monthlyInterest;

            if (i == loanTerm * 12 && remainingBalance > 0) {
                monthlyPayment += remainingBalance;
                monthlyPrincipal = remainingBalance;
            }

            if (i % (paymentFrequency.equals("w") ? 4 :
paymentFrequency.equals("b") ? 2 : 1) == 0) {
                monthlyPayment += additionalMonthlyPayment;
            }

            if (monthlyPayment > remainingBalance + monthlyInterest)
{
                monthlyPayment = remainingBalance + monthlyInterest;
                monthlyPrincipal = remainingBalance;
            }

            remainingBalance -= monthlyPrincipal;
            totalPayment += monthlyPayment;
            totalInterestPaid += monthlyInterest;

            System.out.printf("%d\t\t$%.2f\t\t$%.2f\t\t$%.2f\t\t$%.2
f%n", i, monthlyPayment, monthlyInterest, monthlyPrincipal,
remainingBalance);

            if (remainingBalance <= 0) {
                break;
            }
        }

        System.out.printf("%nTotal payment: $%.2f%n", totalPayment);
        System.out.printf("Total interest paid: $%.2f%n",
totalInterestPaid);

```



```

    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        MoneyLender moneyLender = new MoneyLender();

        System.out.println("Welcome to the Money Lender
Calculator!");

        System.out.print("Please enter the loan amount: ");
        double loanAmount = scanner.nextDouble();
        moneyLender.setLoanAmount(loanAmount);

        System.out.print("Please enter the interest rate (%): ");
        double interestRate = scanner.nextDouble();
        moneyLender.setInterestRate(interestRate);

        System.out.print("Please enter the loan term (years): ");
        double loanTerm = scanner.nextDouble();
        moneyLender.setLoanTerm(loanTerm);

        System.out.print("Please enter the payment frequency (m, w,
or b): ");
        String paymentFrequency = scanner.next();
        moneyLender.setPaymentFrequency(paymentFrequency);

        System.out.print("Please enter any lump sum payment (if
applicable): ");
        double lumpSumPayment = scanner.nextDouble();
        moneyLender.setLumpSumPayment(lumpSumPayment);

        System.out.print("Please enter any additional monthly
payment (if applicable): ");
        double additionalMonthlyPayment = scanner.nextDouble();
        moneyLender.setAdditionalMonthlyPayment(additionalMonthlyPay
ment);

        System.out.println();

        moneyLender.calculateLoan();
    }
}

```

FEATURES OF JAVA:

1) Class: Class is a set of object which shares common characteristics/ behavior and common properties/ attributes. Class is not a real-world entity. It is just a template or blueprint or prototype from which objects are created. Class does not occupy memory. Class is a group of variables of different data types and a group of methods.

2) Methods: The method written by the user or programmer is known as a user defined method. These methods are modified according to the requirement. In this Project various User Defined Methods are defined like setLoanAmount (), setInterestRate(), setLoanTerm(), setPaymentFrequency(), calculateLoan(), etc. which are called In different instances and are performing specific function.

3) Encapsulation: Encapsulation in Java refers to integrating data (variables) and code (methods) into a single unit. In encapsulation, a class's variables are hidden from other classes and can only be accessed by the methods of the class in which they are found.

4) Inheritance: The code extends the functionality of the standard Java class "Scanner" by creating an object of the "Scanner" class to take input from the user.

5) String Manipulation: The code uses the "String" class to manipulate and compare strings, such as checking the value of "paymentFrequency" and formatting output using "printf" with string placeholders.

Github Link:

<https://github.com/neeLabhnagle83/Money-Lender>