

CST 183 Java Programming
Programming Assignment 5 (30 pts)

Purpose

To build a well-documented Java application focusing on Java methods and basic modular programming.

Requirements Write a Java application called LoanCalc.java that will print out information regarding a loan and also capture the output in a file.

The user inputs the following values using interactions with dialog box prompts:

- Principal (amount borrowed - decimal dollars and cents)
- Years of loan (years)
- Interest rate as a percent value (e.g. 4.5, 6.2, etc.)

Initially, validate the user input with use of Java methods.

Method **GetLoanTerm** receives the user choice for the term of the loan (i.e. the years). Acceptable range for loan term is 1 to 30 years.

Method **GetLoanAmount** receives the loan amount. If it is outside the range of 0...\$1 million, your method should return an error.

Method **GetInterestRate** receives the interest rate. Acceptable range is 1% - 8%

For any errors, allow the user to be prompted to try again...use a validation loop!

If the user provides valid input perform the necessary calculations.

How are you going to do this?

Create a java method **CalculatePayment** that will determine the monthly payment for the loan. The following formula defines the monthly payment of the loan.

The formula for calculating the monthly payment for any loan is:

$$PMT = \frac{P}{\left[\frac{1 - (1 + i)^{-n}}{i} \right]}$$

where

P is the loan principal (i.e. the amount borrowed)

i is monthly interest rate (`annual_interest_rate / 12`; expressed as decimal)

N is time (number of monthly payments in total years of loan; i.e. `years * 12`)

To calculate the interest portion of the payment divide the interest rate by 12 (after converting to decimal value (`IR / 100`)). Multiply this value times the remaining balance on the loan.

The remaining balance is always calculated by subtracting the principle portion of the payment.

Finally, and use a method **CreateAmortTable** to generate an amortization table similar to the following: Print to the screen and save in output file.

Month	Interest	Payment	Remaining
1	\$141.67	\$950.00	\$9191.67
2	\$130.22	\$950.00	\$8371.89
.... Etc.			

Make it look nice!

Step 2: Develop Logic Plan

Draw a basic flowchart or write pseudocode that details your logic plan for solving this problem and helps you plan the program. Think about the program flow I've laid out in these instructions.

Step 3: Code It

Use jGRASP or your IDE to write the Java program `loanCalculator.java` that allows the user to perform loan calculations.

1. **Type** in your program code
2. **Compile** your program and correct all the syntax errors
3. **Run** your program and evaluate the output for correctness

Deliverables Submit the Java source code file (the .java file you completed) and the output log file to the **Program 5 Dropbox** within the Delta eLearning System.
Create a hardcopy for turn-in and grading containing the following:

- a) Title Page
- b) Hierarchy chart for program
- c) Source Code (copy and paste from your IDE)
- d) Output file. (copy and paste from your IDE and/or do a screen capture to get the output window).