Student Name: Curtis Foster

Class and Section CIS 2212 103

# Assignment: Comparing Loans

## Problem Description:

Write a program that lets the user enter the loan amount and loan period in number of years and displays the monthly and total payments for each interest rate starting from 5% to 8%, with an increment of 1/8. Here is a sample run:

Loan Amount: 10000

Number of Years: 5

Interest Rate Monthly Payment Total Payment

5% 188.71 11322.74

5.125% 189.28 11357.13

5.25% 189.85 11391.59

...

...

7.875% 202.17 12129.97

8.0% 202.76 12165.83

**Analysis:** (Describe the problem including input and output in your own words.)

This program I would declare local variables to store the interest rate, monthly and total payment, and number of years. Then I would declare loop that iterates through interest rate, but to work around floating-point errors I will do this:

* declare a string that stores a string that uses the format method to set decimal point to three. Then take the double and convert it to string.
* convert it back to double so that I can calculate monthly and total payment.
* display results of interest rate, monthly, and total payments.

**Design:** (Describe the major steps for solving the problem.)

* Import Java Utility Scanner
* Import Java math big decimal and rounding mode.
* Declare Public Class
* Declare Scanner Object
* Prompt User for Loan amount.
* Declare Big Decimal Object
* Prompt User for number years.
* Display labeling for Interest Rate, Monthly, and Total Payment
* Declare For loop set to double starting 5.0 comparing till 8 while iterating 1/8.
* Declare a big decimal object and set the decimal point.
* Declare variables:
  + monthly rate,
    - Declare variable numerator.
    - Declare variable dominator.
  + monthly amount,
  + total amount.
* Display results.

**Coding:** (Submit the archive file)

**Testing:** (Describe how you test this program)

After implanting this concept, I soon found that this work around would not work because the float point errors are still happening with the calculations. So, after doing research I found an answer to this issue with Big Decimal class.

I tested the output monthly rates, the rate percentage, and total amount while referring to sample output. After inputting several inputs, it was quicker to hard code amount and years. As the testing when along, I started to use the debug tool in eclipse to look step by step for errors while parting out parts of the program. What sparked me doing this was I kept getting the wrong output for monthly payments. It was not till I looked through variable names till I found my logic error. In the “numer” and “domer” variables, I put rate instead of monthly payments in the calculation. This happened since I originally named my monthly payments to “monthlyRateBD.”