CITP 190 – Introduction to Java
Project 5

This project requires you to create one project. The project will calculate the monthly payment on a loan.

To receive full credit for this project you must submit the following:

- A detailed design diagram for the program.
- The source code for each program (the .java file) following the coding standards.
- The bytecode for each program (the.class file)
- Proof of the correctness of your output using the test data provided. Please note that you must provide proof for all test data. You may provide additional test data.
- A capture of the output of each program. The output must show all data from your proof.

Submit all files as one (1) ZIP file to the Project 5 Drop Box in the course site.

Grading:

Program design	5 points
Design diagram	5 points
Following course standards	5 points
Proof	3 points
Screen captures	2 points

Important notes:

1. Incorrect calculations will result in a 0 grade for this project.
2. Output that is not presented as shown (including spaces and spelling) will result in a 0 grade for this project.
3. Code that does not follow the standards will result in a 0 grade for this project.
4. Uploading more than one zip file will result in a 0 grade for this project.
5. Not using all the test data provided will result in a 0 grade for this project.
6. Using a `continue` statement or misusing a `break` statement will result in a 0 grade for this project.

Project 5: Calculate the monthly payment on a loan

Console

```
Welcome to the Loan Calculator

DATA ENTRY
Enter loan amount:          ten
Error! Invalid decimal value.

Enter loan amount:          -1
Error! Number must be greater than 0.0

Enter loan amount:          100000000000
Error! Number must be less than 1000000.0

Enter loan amount:          500000
Enter yearly interest rate: 5.6
Enter number of years:      thirty
Error! Invalid integer value.

Enter number of years:      -1
Error! Number must be greater than 0

Enter number of years:      100

Error! Number must be less than 100
Enter number of years:      30

RESULTS
Loan amount:          $500,000.00
Yearly interest rate: 5.6%
Number of years:      30
Monthly payment:      $2,870.39

Continue? (y/n):
Error! This entry is required. Try again.

Continue? (y/n): x
Error! Entry must be 'y' or 'n'. Try again.

Continue? (y/n): n
```

Operation

- The Data Entry section prompts the user to enter values for the loan amount, yearly interest rate, and number of years. If the user doesn't enter data that's valid, this section displays an appropriate error message and prompts the user again.
- The Results section displays a formatted version of the user's entries as well as the formatted result of the calculation.
- The application prompts the user to continue.

Specifications

- The formula for calculating monthly payment is:

```
double monthlyPayment = loanAmount * monthlyInterestRate/
                        (1 - 1/Math.pow(1 + monthlyInterestRate, months));
```

- The application should accept decimal entries for the loan amount and interest rate entries.
- The application should only accept integer values for the years field.
- The application should only accept integer and decimal values within the following ranges:

```
                        Greater     Less
                        Than        Than
Loan amount:            0           1,000,000
Yearly interest rate:   0           20
Years:                  0           100
```

- The application should only accept a value of "y" or "n" at the Continue prompt.
- If the user enters invalid data, the application should display an appropriate error message and prompt the user again until the user enters valid data.
- The code that's used to validate data should be stored in separate methods. For example:

```
public static double getDoubleWithinRange(Scanner sc, String prompt, double min, double max)

public static int getIntWithinRange(Scanner sc, String prompt, int min, int max)
```

Test Data

Loan Amount	Interest Rate	Years
50000	7.5%	20
-500, 1500	12.5%	2
2000000, 20000	21.3%, 2.13%	5
1000000	-5%, 5.5%	-2, 200, 20

Note: This test data includes error data. For example -500 is not a valid loan amount. Whenever more than one value is included, the first value(s) should produce error message(s). The last value should be a good value. In your proof, note which values are error data and perform the calculations on the good data.