Project 5 (50 points)
Due MONDAY, October 24 by midnight

Assignment Description:

To simulate an actual real world programming project, you are given the following *general description* of what is needed by *Big12 Bank Midwest*:

> *A bank needs software (i.e. a program) that can be used to figure monthly payment and the total amount to be paid on various mortgage (house) loans. Program should be flexible enough to either allow a bank officer to enter in all the input or use a special advertised promotion where the customer can get a $100,000 house loan for 15 years at the annual rate of 5.5%. The program should provide some type of menu that allows the bank officer to choose between the two. Program should continue until the user decides to end the input.*

Develop an object-oriented solution to the program requested.

Implementation Requirements:

Your project will contain **two** classes: ***Mortgage*** and ***Proj5***.

Remember that your driver program (***Proj5***) must contain only a main method (unless doing the extra credit below) that includes displaying the menu, getting and validating the user's choice and input on unique loans, object declarations and an outline of method calls to methods defined in your *Mortgage* class.

1) **Valid interest rates are between 1%-9% (inclusive)**
2) **Valid terms are 5-50 years (inclusive)**
3) **Valid amounts are $50,000 to 1 million (inclusive)**

Mortgage will be a general class that represents a loan mortgage (i.e. a house loan) and will be used to create *Mortgage* objects. At a minimum, it will contain private instance variables for the three values needed to calculate a mortgage, one to hold the monthly payment, and one to hold the total amount paid on the loan. It will also contain a minimum of two constructors used to initialize their values. It must also define the following methods (to maximize points earned, use method signatures given):

```
// Calculate and set the monthly payment for the loan
   public void setMonthlyPayment()

// Calculate and set the total payment for the loan
   public void setTotalPayment()

// Display formatted output
   public void displayInfo()
```

- Format all currency values with '$'-signs, commas, and two values after the decimal. (Hint: use the *DecimalFormat* class to format numeric values)

<u>For Extra Credit (+5)</u>, add the following methods to your *Proj5* class…

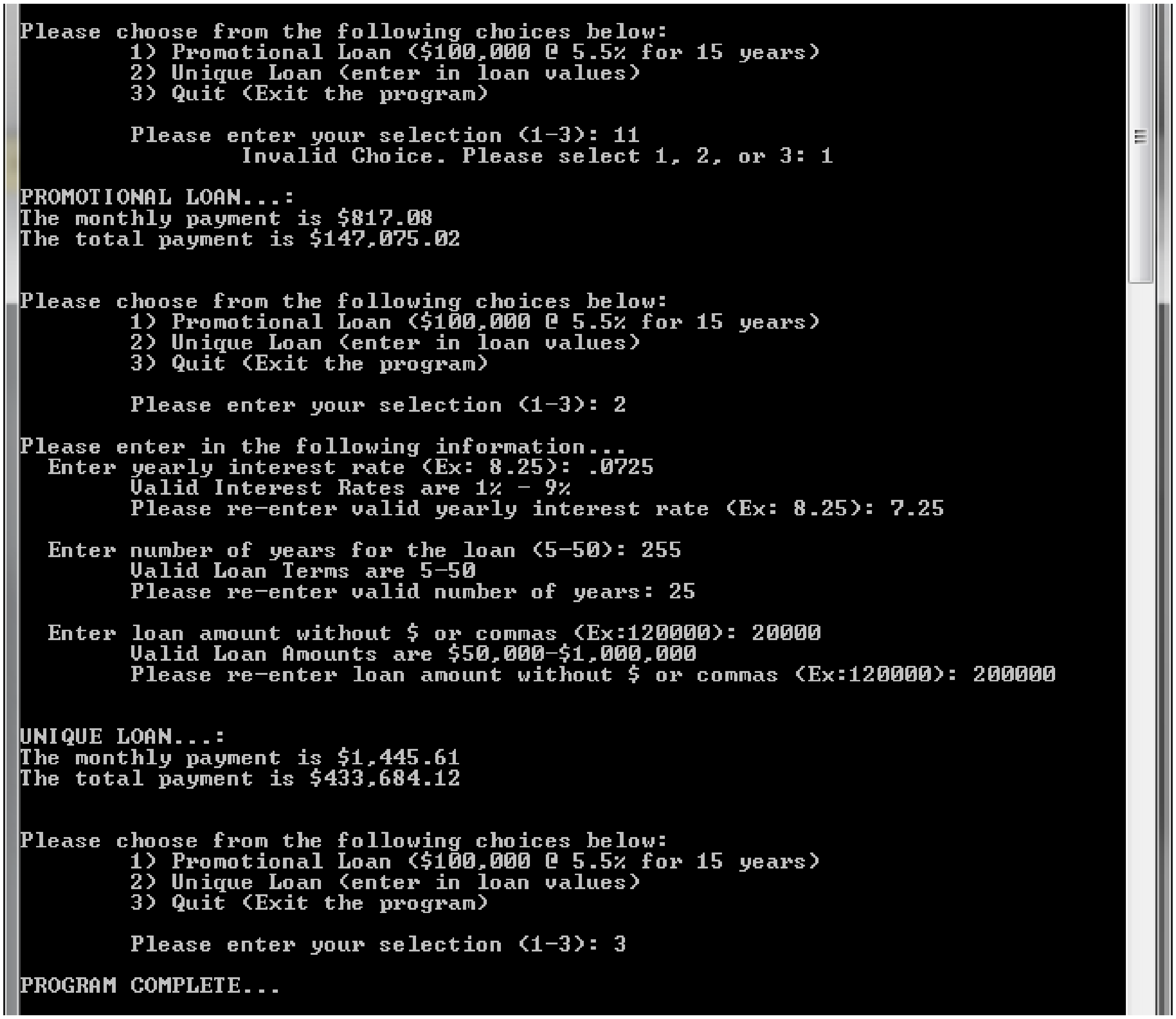
// reads in and validates interest rate … valid interest rates are between 1%-9% (inclusive)
getRate()

// reads in and validates length of the loan … valid terms are 5-50 years (inclusive)
getTerm()

// reads in and validates loan amount … valid amounts are $50,000 to 1 million (inclusive)
getAmount()

Note: Extra Credit available only if the rest of the program is completed correctly.

A ***<u>possible</u>*** execution of your program might look like the following…



```
Please choose from the following choices below:
        1) Promotional Loan ($100,000 @ 5.5% for 15 years)
        2) Unique Loan (enter in loan values)
        3) Quit (Exit the program)

        Please enter your selection (1-3): 11
                Invalid Choice. Please select 1, 2, or 3: 1

PROMOTIONAL LOAN...:
The monthly payment is $817.08
The total payment is $147,075.02


Please choose from the following choices below:
        1) Promotional Loan ($100,000 @ 5.5% for 15 years)
        2) Unique Loan (enter in loan values)
        3) Quit (Exit the program)

        Please enter your selection (1-3): 2

Please enter in the following information...
  Enter yearly interest rate (Ex: 8.25): .0725
        Valid Interest Rates are 1% - 9%
        Please re-enter valid yearly interest rate (Ex: 8.25): 7.25

  Enter number of years for the loan (5-50): 255
        Valid Loan Terms are 5-50
        Please re-enter valid number of years: 25

  Enter loan amount without $ or commas (Ex:120000): 20000
        Valid Loan Amounts are $50,000-$1,000,000
        Please re-enter loan amount without $ or commas (Ex:120000): 200000


UNIQUE LOAN...:
The monthly payment is $1,445.61
The total payment is $433,684.12


Please choose from the following choices below:
        1) Promotional Loan ($100,000 @ 5.5% for 15 years)
        2) Unique Loan (enter in loan values)
        3) Quit (Exit the program)

        Please enter your selection (1-3): 3

PROGRAM COMPLETE...
```

Documentation:
You should put a description of the project at the top of the file **and at the top of each method**. Please use this template for the top of the file:

```
/**
 * (description of the project)
 *
 * @author (your name)
 * @version (which number project this is)
 */
```

Please use this template for the top of each method:

```
/**
 * (description of the method)
 *
 * @param (describe first parameter)
 * @param (describe second parameter)
 * (list all parameters, one per line)
 * @return (describe what is being returned)
 */
```

Submission:
To submit your project, first create a folder called proj5, and move your completed *Proj5.java* **and** *Mortgage.java* files into that folder. Then, right-click on that folder and select "Send To → Compressed (zipped) folder". This will create the file proj5.zip.

Go to "Files and Content->Modules->Submit Projects Here" on K-State Online. Select your lab time and upload the proj5.zip file. **Put your name and Project 5 in the description box.**

Grading:

Programs that do not compile will receive a grade of 0.

Requirement	**Points**
Proj5 class contains only a main method (unless doing extra credit) that includes displaying the menu, getting and validating the user's choice, object declarations and an outline of method calls to methods defined in *Mortgage* class	**8**
Mortgage class contains required private variables, two constructors and required methods with given method signatures	**8**
Prompts user for a choice, validates and creates object by calling the proper constructor	**5**
Properly Validates all input on *unique* loans	**6**
Prints result by calling a method – value format exactly matches example ('$'-sign, Comma, two decimal places on all values)	**6**
setMonthlyPayment is correct	**4**
setTotalPayment is correct	**4**
displayInfo is correct	**4**
Loops until user chooses to quit	**2**
Documentation/naming/submission	**3**
Extra Credit – 3 methods correctly added to *Proj5* *(***available only if the rest of the program is completed correctly***)*	**+5**
Total	**50**