Project #6 – Amy's Auto Loan Payments

1 Problem Overview

You are a consultant doing work for Amy's Autos, a small car dealership. Amy wants a program that lets her staff enter basic loan data (amount, interest rate) and show the potential buyer various financing options, including a report showing a monthly payment breakdown.

2 User Interface

2.1 Initial Input

The user is first asked for the loan amount and annual interest rate:

```
Enter loan amount, example 10000:  20000
Enter annual interest rate, example 2.9:  2.9
```

If the user enters out-of-range data, prompt them again until they get it right. Loan amounts must not be negative. Interest rates must not be negative.

These inputs are used for *all* further calculations.

2.2 Input Menu

The user is then shown a menu offering various loan terms from which they can choose:

```
--------------------------------
Amy's Auto - Loan Report Menu
--------------------------------
1.  12-month loan
2.  24-month loan
3.  36-month loan
4.  48-month loan
5.  60-month loan
0.  EXIT

Choice:
```

2.3 Output

Once the user chooses a loan term option from the menu, they get a nicely formatted report showing one line per payment the buyer will make (abbreviated for space's sake, here):

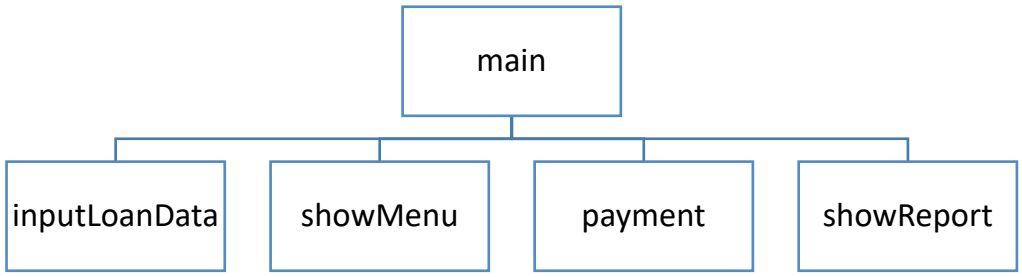
```
Pmt#      PmtAmt       Int        Princ      Balance
----   ---------     -----      ------     --------
   1    1,692.96     48.33    1,644.63   18,355.37
   2    1,692.96     44.36    1,648.60   16,706.77
...
  11    1,692.96      8.15    1,684.81    1,688.90
  12    1,692.98      4.08    1,688.90        0.00
        ---------
        20,315.54

Press <Enter> to continue
```

Once the user presses <Enter>, the menu of term options is once again displayed.

3 Functions

3.1 Call Hierarchy



3.2 Specific Functions

Function Name	Parameters	Returns	Description
main	none	none	• Calls inputLoanData to gather loan info • Sets up a loop that calls functions that display the menu and chosen reports • The loop ends when it sees the menu return a loan term of 0 months
inputLoanData	none	Loan amount[1], Interest rate per period	• Asks the user for the basic loan info Implement validation LOOPS to ensure valid data is collected/returned.
showMenu	none	Number of periods	• In a loop, displays the loan term menu and asks for a user choice • If the user makes a valid choice, it returns the number of periods • If the user makes an invalid choice, the menu simply displays again • If the user chooses the Exit option (choice 0), it returns a loan period of 0 months as a signal to the calling function that the user wants out
payment	PV, Rate per period[2], Number of periods[3]	Payment amount	• Calculates the payment as shown in the Calculations section below
showReport	PV, Rate per period, Number of periods, Payment amount	None	• Displays an attractive, aligned report showing the payment breakdown (which it calculates as shown below) • After the report, displays a message asking the user to press <Enter> to continue

[1] This is also used as PV, the present value of the loan.
[2] This is also known as the periodic interest rate.
[3] We're talking about monthly payments, so for our purposes "periods" = "months" throughout this project.

4 Calculations

Here are the key calculations you'll need to use:

- **Payment Amount** is calculated using this formula: where PV is Present Value of the loan, r = rate per period (see Hints below), and n is the number of periods. Round to the nearest penny.

$$P = \frac{\mathrm{r(PV)}}{1 - (1 + r)^{-n}}$$

  Adjust the last payment, principal, and balance to take care of any small amount left after the last payment has been calculated (note the last line of the report above). Sum the payments and display that result at the bottom of the report (as shown above).
- **Interest** for a specific monthly payment is calculated by taking the current loan balance and multiplying by the rate per period, rounded to the nearest penny.
- **Principal** is calculated by taking the Payment Amount and subtracting the **Interest** portion.
- **Balance** is calculated by taking the old loan balance and subtracting the **Principal** amount.

5 Code Specifications

- At the **bottom** of the program you should have a call to main().
- Include header comments at the top of each file. Include your name, the date, and a brief description of what the program does.
- Include comments for each section saying what's going on in the lines of code below.
- Use comments elsewhere as you think they help guide the reader. Don't overdo, though! Not every line needs a comment; think about describing a block of related code.
- Use blank lines to separate sections and provide visual "breathing room."
- Use descriptive variable names.

6 Hints

- The report needs to look *nice*. Note the formatting in the sample; replicate that or do better if you're up for the challenge. *Don't use tabs and spaces for alignment*; you know better, now.
- The user-entered interest rate is typed in human-readable notation; you'll need to alter it before using it in loan calculations, e.g., converting 2.9 to .029 for math purposes. Also note that this is an *annual rate*; that's not what the formulas may want, so another operation will also be needed to get the rate per period.
- You should end up with two *while* loops and one *for* loop in the program.
- Your report function needs something like a "running total" as the loan balance keeps decreasing as payments are made. It also needs an actual running total (of payments).
- Use *exactly* the call hierarchy shown (though you can create helper functions); do not *chain*! Helper functions are allowed.

7 Testing

- Develop an appropriate number of test cases.
- Document your testing and results in comments at the bottom of the program as shown below.

8 Summary

At the bottom of your program, add comments that answer these questions:

- How did you approach this assignment? Where did you get stuck, and how did you get unstuck?
- How did you test your program? What doesn't work as you'd like, perhaps things that you'd like to fix as you learn more?
- What did you learn from this assignment? What will you do differently on the next project?

9 Grading Matrix

Area	**Percent**
Menu display	10
User input	10
User input range validation	20
Menu/output looping	20
Calculations correct	20
Test cases	10
Internal documentation	5
Summary report	5
Total	**100**