

Programming Fundamentals

CS1336

Assignment #3

Assignment #3 – Calculations Continued

Introduction

Your third programming assignment will consist of two small C++ programs. Each program will be independent of the other program. Each one should compile correctly and produce the specified output.

Please note that each of the programs should comply with the commenting and formatting rules we discussed in class. For example, there should be a header for the whole program that gives the author's name, class name, date, and description. End braces should be commented, and there are alignment and indenting requirements as discussed. Please ask if you have any questions.

Program #1 – Using a `string` variable

For the first problem, please implement a modified version of Problem 4 on page 144 of the text (9th Edition). A scan of the problem is provided below.

The basic problem asks you to calculate the average rainfall for three months. The program should ask the user to enter the name of each month, such as June or July, and the amount of rain (in inches) that fell each month in a set of three consecutive pairs of questions:

```
Please enter the name of month 1: <August>
Please enter the rainfall for <month entered>: 5.3
```

```
Please enter the name of month 2: <September>
Please enter the rainfall for <month entered>: 8.425
```

followed by one more pair for month 3. Once the data have been entered, the program should then display a report as follows:

```
The average rainfall for June, July, and August was 6.72 inches.
```

where, of course, the months given were the months entered by the user and the rainfall is the result calculated by your program. Your program should work for any three months typed in by the user, no matter what they are.

Note that the rainfall average is printed to two decimal places this time.

In this program we are asking the user for string data, not just numeric input. To accomplish that in a C++ program, we can use the `string` class. Simply declare three variables of type `string` to receive the month name input from `cin`. Don't forget to include the `<string>` header file.

Finally, to get some practice using the C++ `string` class, let's make use of some of the `string` member functions. For a final report, have the program print out the length of each of the three months entered. For example:

```
The length of "August" is:      6 characters
The length of "September" is:  9 characters
The length of "November" is:   8 characters
```

where “August” is one of the months entered, etc. Note that your program should put double quotes around the month in the output.

4. Average Rainfall

Write a program that calculates the average rainfall for three months. The program should ask the user to enter the name of each month, such as June or July, and the amount of rain (in inches) that fell each month. The program should display a message similar to the following:

```
The average rainfall for June, July, and August is 6.72 inches.
```

Program #2 – Mortgage Calculation

For a more complex calculation program, please implement Problem 19 on page 147 of our text (9th Edition).

You may use the `pow()` function in the `cmath` header. This function has the following prototype:

```
double pow (double base, double power);
```

The “base” is the number that is being taken to some power, and the “power” is the power to which the base is taken. The function will return a `double` as the result.

Please refer to pp 95-97 in our textbook for a description of this function and how to use it.

Your program should prompt for each of the three values indicated in the problem (the loan amount, the annual interest rate, and the number of payments), perform the calculations, and then print a report similar to the one in the book. Note that the calculations are done based on the monthly interest rate. After your program asks the user for the annual interest rate, you must convert it to a monthly rate by dividing by 12.

Here is a scan from the book to help some of you who may not have the text.

17. Monthly Payments

The monthly payment on a loan may be calculated by the following formula:

$$\text{Payment} = \frac{\text{Rate} * (1 + \text{Rate})^N}{((1 + \text{Rate})^N - 1)} * \text{L}$$

Rate is the monthly interest rate, which is the annual interest rate divided by 12. (12% annual interest would be 1 percent monthly interest.) N is the number of payments and L is the amount of the loan. Write a program that asks for these values and displays a report similar to

Loan Amount:	\$ 10000.00
Monthly Interest Rate:	1%
Number of Payments:	36
Monthly Payment:	\$ 332.14
Amount Paid Back:	\$ 11957.15
Interest Paid:	\$ 1957.15

Let's make one change to the way the interest rate is printed out. In the real world, interest rates almost always reported with decimal values, and not an integer as depicted here. So let's print that value out to two decimal places. Therefore, if the monthly interest rate is calculated to be 1%, it should be displayed as 1.00%.

For a practical example, let's say the user enters 6.35% as the annual interest rate. Then the monthly interest rate would be $6.35 / 12 = 0.529166667\%$. We should report that to two decimal places as 0.53%, with the percent sign lined up on the rightmost edge of the report.

Don't forget that that number is a percentage, so before you use it in the formula you have to divide it by 100. Therefore, the actual rate used in the formula would be 0.0052916667 in this particular example.