

# Case Study: Loan Amortization

The Central Mountain Credit Union uses a central data processing system where each branch accesses a networked server at the Credit Union's main office. The server has a tendency to slow down during certain times, when many users are on the system. Because of this, the loan officer at one of the branch offices has asked you to write a loan amortization program to run locally, on her desktop PC.

## Calculations

The credit union uses the following formula to calculate the monthly payment of a loan:

$$\text{Payment} = \frac{\text{Loan} * \text{Rate}/12 * \text{Term}}{\text{Term} - 1}$$

Where:

Loan = the amount of the loan

Rate = the annual interest rate

Term =  $(1 + R/12)^{Y*12}$

## Report Requirements

The report produced by the program should show the monthly payment and print four columns for each month in the loan period:

Month number  
Interest  
Principal  
Balance

The following report may be used as a model. It shows all the required information on a one-year, \$5,000 loan at 12.9 percent annual interest.

Monthly payment: \$446.35

Month	Interest	Principal	Balance
1	53.75	392.60	4607.40
2	49.53	396.82	4210.58
3	45.26	401.09	3809.49
4	40.95	405.40	3404.09
5	36.59	409.76	2994.33
6	32.19	414.16	2580.17
7	27.74	418.62	2161.55
8	23.24	423.12	1738.44
9	18.69	427.66	1310.77
10	14.09	432.26	878.51
11	9.44	436.91	441.60
12	4.75	441.60	0.00

Variables

Table 1 lists the variables needed in the program.

Table 1

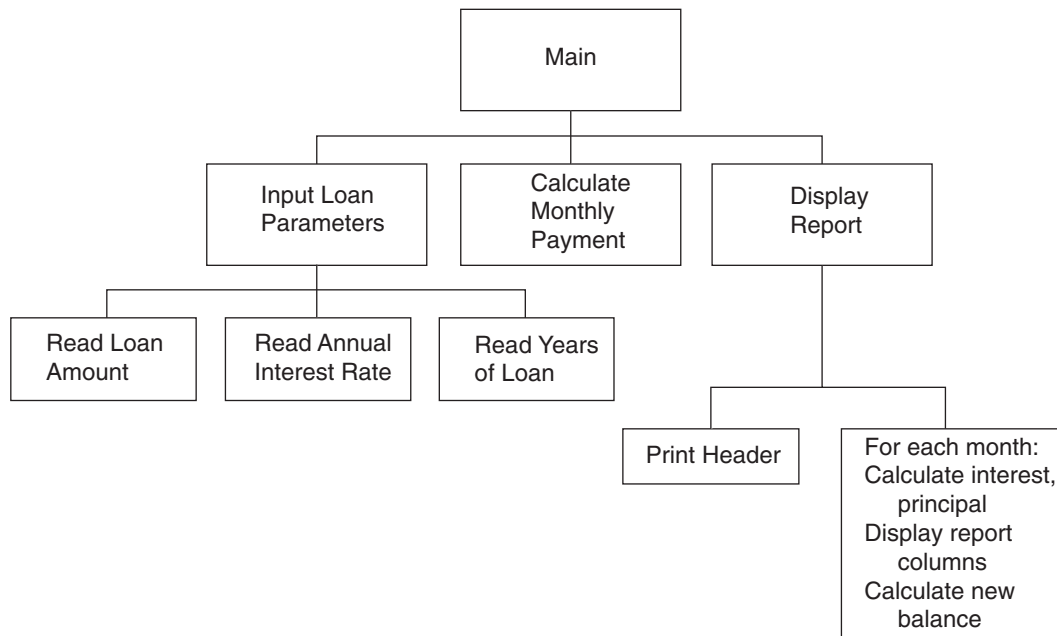
Variable	Description
loan	A double. Holds the loan amount.
rate	A double. Holds the annual interest rate.
years	A double. Holds the number of years of the loan.
balance	A double. Holds the monthly balance.
term	A double. Used in the payment calculation.
payment	A double. Holds the monthly payment.

Program Design

Figure 1 shows a hierarchy chart for the program.

The detail of the program can be expanded in pseudocode:

```
Ask user to input the loan amount.
Ask user to input the annual interest rate.
Ask user to enter the number of years of the loan.
Calculate the monthly payment.
Print the report header.
For each month in the loan period
    Calculate the monthly interest.
    Calculate the principal.
    Display the month, payment, interest, principal, and balance.
    Calculate the new balance.
End For
```

**Figure 1**

Program 5-19 lists the final C++ code:

### Program 5-19

```

1  // This program produces a loan amortization chart for the
2  // Central Mountain Credit Union.
3  #include <iostream>
4  #include <iomanip>
5  #include <cmath> // For pow function
6  using namespace std;
7
8  int main()
9  {
10     const int MONTHS = 12; // Months per year
11     double loan,           // Loan amount
12           rate,            // Annual interest rate
13           years,          // Years of loan
14           balance,         // Monthly balance
15           term,            // Used to calculate payment
16           payment;         // Monthly payment
17
18     // Ask user for input.
19     cout << "Loan amount: $";
20     cin >> loan;
21     cout << "Annual interest rate: ";
22     cin >> rate;
23     cout << "Years of loan: ";

```

*(program continues)*

**Program 5-19***(continued)*

```
24     cin >> years;
25
26     // Calculate monthly payment.
27     term = pow((1 + rate / MONTHS), MONTHS * years);
28     payment = (loan * rate / MONTHS * term) / (term - 1.0);
29
30     // Display monthly payment.
31     cout << fixed << showpoint << setprecision(2);
32     cout << "Monthly payment: $" << payment << endl;
33
34     // Display report header.
35     cout << endl;
36     cout << setw(5) << "Month";
37     cout << setw(10) << "Interest";
38     cout << setw(10) << "Principal";
39     cout << setw(10) << "Balance" << endl;
40     cout << "-----\n";
41
42     // Produce a listing for each month.
43     balance = loan;
44     int numPayments = MONTHS * years;
45     for (int month = 1; month <= numPayments; month++)
46     {
47         double minterest, principal;
48         // Calculate monthly interest
49         minterest = rate / MONTHS * balance;
50         if (month != numPayments)
51             principal = payment - minterest;
52
53         else // If this is the last month
54         {
55             principal = balance;
56             payment = balance + minterest;
57         }
58
59         // Calculate the new loan balance.
60         balance -= principal;
61
62         // Display payment figures
63         cout << setw(4) << month;
64         cout << setw(10) << minterest;
65         cout << setw(10) << principal;
66         cout << setw(10) << balance << endl;
67     }
68     return 0;
69 }
```

**Program 5-19***(continued)***Program Output with Example Input Shown in Bold**Loan amount: **\$2500 [Enter]**Annual interest rate: **.08 [Enter]**Years of loan: **2 [Enter]**

Monthly payment: \$113.07

Month	Interest	Principal	Balance
-----			
1	16.67	96.40	2403.60
2	16.02	97.04	2306.55
3	15.38	97.69	2208.86
4	14.73	98.34	2110.52
5	14.07	99.00	2011.52
6	13.41	99.66	1911.86
7	12.75	100.32	1811.54
8	12.08	100.99	1710.55
9	11.40	101.66	1608.89
10	10.73	102.34	1506.54
11	10.04	103.02	1403.52
12	9.36	103.71	1299.81
13	8.67	104.40	1195.40
14	7.97	105.10	1090.31
15	7.27	105.80	984.51
16	6.56	106.50	878.00
17	5.85	107.21	770.79
18	5.14	107.93	662.86
19	4.42	108.65	554.21
20	3.69	109.37	444.83
21	2.97	110.10	334.73
22	2.23	110.84	223.90
23	1.49	111.58	112.32
24	0.75	112.32	0.00