```
/* FILE NAME: ……….
 * AUTHOR: Solution Briefing
 * See our syllabus for a good comment
 */

#include <iostream>
using namespace std;

int main() {
	// VARIABLE INITIALIZATION

	Properly define your variables……..
	          ………………….. 

	// CURRENCY FORMATTING
	cout.setf(ios::fixed);
	cout.setf(ios::showpoint);
	cout.precision(2);

	// USER INPUT
	// NOTE: For valid input, the loan, interest, and monthly payment must
	// be positive. The monthly payment must also be large enough to
	// terminate the loan.
	cout << "\nLoan Amount: ";
	cin >> loan;

	Your program will not move forward until a positive load is entered


	cout << "Interest Rate (% per year): ";
	cin >> interestRate;

	Your program will not move forward until a positive interest rate is entered


	// GET PROPER INTEREST RATES FOR CALCULATIONS
	interestRate /= 12;
	interestRateC = interestRate / 100;
	cout << "Monthly Payments: ";
	cin >> monthlyPaid;

	Your program will not move forward until a monthly payment is sufficient


	cout << endl;

	// AMORTIZATION TABLE
	cout << "*****************************************************************\n"
	     << "\tAmortization Table\n"
	     << "*****************************************************************\n"
	     << "Month\tBalance\t\tPayment\tRate\tInterest\tPrincipal\n";

	// LOOP TO FILL TABLE
	while (loan > 0) {
		if (currentMonth == 0) {
			cout << currentMonth++ << "\t$" << loan;
```

```
              if (loan < 1000) cout << "\t"; // Formatting MAGIC
                     cout << "\t" << "N/A\tN/A\tN/A\t\tN/A\n";
              }
              else {

              Properly calculate and display “montlypaid” and “principal” when
              (1) loan * (1 + interestRateC) < monthlyPaid
              and (2) loan * (1 + interestRateC) >= monthlyPaid

       }
       cout << "****************************************************************\n";
       cout << "\nIt takes " << --currentMonth << " months to pay off "
            << "the loan.\n"
            << "Total interest paid is: $" << interestTotal;
       cout << endl << endl;
       return 0;
}
```