Learning Objectives:

1. Review Unix commands from last lab:
   a. man ls
   b. tar xvf lab07.tar
2. Use the 6 relational operators correctly in your programs
3. Understand the difference between:

If (x==1) {
  cout << "1" << endl;
  cout << "2" << endl;
}
	If (x==1)
	  cout << "1" << endl;
	  cout << "2" << endl;

4. Avoid some common coding pitfalls
5. Practice using the while loop for coding
6. Apply the "magic formula" for formatting numbers (Ch. 1.3)

```
cout.setf(ios::fixed);
cout.setf(ios::showpoint);
cout.precision(2);
```

7. Use **%** in vi text editor to match the braces

Pitfalls

Using = in Place of ==. According to the textbook, even the very best of programmers fall into this error.

For example, the following will check to see if x is equal to 1:

```
if ( x == 1 )
```

But if you write:

```
if ( x = 1 )
```

The value of 1 is transferred to the variable **x** on the left-hand side of the assignment and then the condition inside the if-statement will always be true.

The program will run and produce incorrect results.

Check out pitfall1.cpp and understand the logical errors in the program.

Extra Semicolon.

This error occurs when a semicolon is added to the end of an if statement or for loop, e.g.:

```
if(x == 12);
  x = 0;
```

After this code segment, the variable `x` has the value 0. The error is the semicolon at the end of the line with the `if`. The `if` statement expects to see a single statement after the closing parenthesis. The semicolon produces a null statement between the close parenthesis and the semicolon. This makes the next statement (`x = 0;`) out of the scope of control of the `if`.

The extra semicolon problem can cause an infinite loop, for example,

```
x = 10;
while(x > 0);
{
  cout << x << endl;
  x--;
}
```

A program that has this loop in it will hang. As before, the extraneous semicolon after the `while` causes the problem. To stop such runaway programs may require killing the process or hitting control-C.

Installment Loan Time (Chapter 2 question 2 in the textbook)

You have just purchased a stereo system that cost $1,000 on the following credit plan:

- no down payment
- an interest rate of 18% per year (1.5% per month)
- and monthly payments of $50

The monthly payment of $50 is used to pay the interest, and whatever is left is used to pay part of the remaining debt. Hence, the first month you pay 1.5% of $1,000 in interest. That is $15 in interest. The remaining $35 is deducted from your debt, which leaves you with a debt of $965.00. The next month you pay interest of 1.5% of $965.00, which is $14.48. Hence, you can deduct $35.52 (which is $50 − $14.48) from the amount you owe.

Write a program that will tell you how many months it will take you to pay off the loan, as well as the total amount of interest paid over the life of the loan. **Use a while loop** to calculate the amount of interest and the size of the debt after each month.

Use a variable to count the number of loop iterations and hence the number of months until the debt is zero. You may want to use other variables as well. The last payment may be less than $50 if the debt is small, but do not forget the interest. If you owe $50, then your monthly payment of $50 will not pay off your debt, although it will come close. One month's interest on $50 is only 75 cents.

A typical run is:

```
months      interest    principal
1           15.00       965.00
2           14.47       929.48
3           13.94       893.42
4           13.40       856.82
. . .
21          2.86        143.53
22          2.15        95.68
23          1.44        47.12
number of payments = 24 last months interest = 0.71 last payment = 47.83
```