The formula for determining the monthly payment \( p \) for a mortgage is:

\[
p = \frac{m(1 + \left(\frac{r}{12}\right)^{12n} - 1)\frac{r}{12}}{12}
\]

where \( m \) is the amount of the loan, \( r \) is the annual interest rate, and \( n \) is the number of years in the mortgage. For example, a 30-year loan for $105,000 at 11.5\% (i.e., \( r = .115 \)) results in a monthly payment of $1039.81 rounded to the nearest cent.

Write a program that obtains (from the user) the amount, interest rate and number of years of the loan, then calculates and prints out the monthly payment.

An example run, with user entries shown here in \textbf{BOLD ITALIC} (for clarity only):

Enter loan amount : \textbf{105000.00}
Enter annual interest rate (e.g., 0.08 for 8\%) : \textbf{0.115}
Enter number of years : \textbf{30}

The monthly payment is $1039.81

A function you will need is \texttt{pow( x, p )}, which returns \( x \) to the \( p \)th power. To get access to it, you need to put the line \texttt{#include <cmath>} at the top of your program. The variables \( x \) and \( p \) in the function must both be of type \texttt{double}, and the returned value is also of type \texttt{double}. You can't mix types in the call to \texttt{pow( )}.

An example call:

double \( x \), \( y \);
    (statements that set \( x \) to some value...)
    \( y = 1.0 + \text{pow}( x, 1.0/3.0 ) \);

This places one more than the cube root of \( x \) into \( y \).

Round the result to 2 decimal places following the model in kilos.cpp. By adding the line \texttt{#include <iomanip>} to your program, you can use the showpoint, fixed and setprecision capabilities. Test your program with the values given above, and by varying the values and verifying the output is reasonable. Include tests with a higher and lower loan amount, interest rate, and term (holding the other two values the same), as well as tests with two or all three of the input values changed.