14. //THIS IS JUST A STUB.
    double rainProb(double pressure,
                      double humidity, double temp)
    {
        return 0.25;  //Not correct,
                     //but good enough for some testing.
    }

MyProgrammingLab

Programming Projects

Visit www.myprogramminglab.com to complete select exercises online and get instant feedback.

1. Write a program that converts from 24-hour notation to 12-hour notation. For example, it should convert 14:25 to 2:25 P.M. The input is given as two integers. There should be at least three functions: one for input, one to do the conversion, and one for output. Record the A.M./P.M. information as a value of type char, 'A' for A.M. and 'P' for P.M. Thus, the function for doing the conversions will have a call-by-reference formal parameter of type char to record whether it is A.M. or P.M. (The function will have other parameters as well.) Include a loop that lets the user repeat this computation for new input values again and again until the user says he or she wants to end the program.

2. The area of an arbitrary triangle can be computed using the formula
   \[ \text{Area} = \sqrt{s(s-a)(s-b)(s-c)} \]
where a, b, and c are the lengths of the sides, and s is the semiperimeter.
   \[ s = (a + b + c)/2 \]
Write a void function that uses five parameters: three value parameters that provide the lengths of the edges, and two reference parameters that compute the area and perimeter (not the semiperimeter). Make your function robust. Note that not all combinations of a, b, and c produce a triangle. Your function should produce correct results for legal data and reasonable results for illegal combinations.

3. Write a program that tells what coins to give out for any amount of change from 1 cent to 99 cents. For example, if the amount is 86 cents, the output would be something like the following:
   86 cents can be given as
   3 quarter(s) 1 dime(s) and 1 penny(pennies)
Use coin denominations of 25 cents (quarters), 10 cents (dimes), and 1 cent (pennies). Do not use nickel and half-dollar coins. Your program will use the following function (among others):
   void computeCoin(int coinValue, int& number, int& amountLeft);
   //Precondition: 0 < coinValue < 100; 0 <= amountLeft < 100.
   //Postcondition: number has been set equal to the maximum number
   //of coins of denomination coinValue cents that can be obtained
//from amountLeft cents. amountLeft has been decreased by the
//value of the coins, that is, decreased by number*coinValue.

For example, suppose the value of the variable amountLeft is 86. Then, after the
following call, the value of number will be 3 and the value of amountLeft will be 11
(because if you take three quarters from 86 cents, that leaves 11 cents):

computeCoins(25, number, amountLeft);

Include a loop that lets the user repeat this computation for new input values until
the user says he or she wants to end the program. (Hint: Use integer division and
the % operator to implement this function.)

4. Write a program that will read in a length in feet and inches and output the equiva-
   lent length in meters and centimeters. Use at least three functions: one for input,
   one or more for calculating, and one for output. Include a loop that lets the user
   repeat this computation for new input values until the user says he or she wants to
   end the program. There are 0.3048 meters in a foot, 100 centimeters in a meter,
   and 12 inches in a foot.

5. Write a program like that of the previous exercise that converts from meters and
   centimeters into feet and inches. Use functions for the subtasks.

6. (You should do the previous two programming projects before doing this one.)
   Write a program that combines the functions in the previous two programming
   projects. The program asks the user if he or she wants to convert from feet and
   inches to meters and centimeters or from meters and centimeters to feet and inches.
   The program then performs the desired conversion. Have the user respond by typ-
   ing the integer 1 for one type of conversion and 2 for the other conversion. The
   program reads the user’s answer and then executes an if-else statement. Each
   branch of the if-else statement will be a function call. The two functions called
   in the if-else statement will have function definitions that are very similar to
   the programs for the previous two programming projects. Thus, they will be fairly
   complicated function definitions that call other functions. Include a loop that lets
   the user repeat this computation for new input values until the user says he or she
   wants to end the program.

7. Write a program that will read in a weight in pounds and ounces and will output
   the equivalent weight in kilograms and grams. Use at least three functions: one for
   input, one or more for calculating, and one for output. Include a loop that lets
   the user repeat this computation for new input values until the user says he or she
   wants to end the program. There are 2.2046 pounds in a kilogram, 1000 grams in
   a kilogram, and 16 ounces in a pound.

8. Write a program like that of the previous exercise that converts from kilograms and
   grams into pounds and ounces. Use functions for the subtasks.

9. (You should do the previous two programming projects before doing this one.)
   Write a program that combines the functions of the previous two programming
   projects. The program asks the user if he or she wants to convert from pounds
   and ounces to kilograms and grams or from kilograms and grams to pounds and
   ounces. The program then performs the desired conversion. Have the user respond
   by typing the integer 1 for one type of conversion and 2 for the other. The program