Project Three
Paint Project

A remodeling company needs a program to determine the cost of painting a room. Write a C++ program that lets the user enter the length and width of a room in feet as integers, and then calculates the total square footage of the room, the number of cans of paint rounded to the next whole number, and the cost of the paint. The height of the room is 8 feet, and the price of a can of paint is 12.99. A one-gallon can of paint covers 250 square feet.

You do not need to factor in the area for doors and windows, but include the ceiling. There are a number of ways to solve this problem. You may want to check the description for the *ceil* and *floor* functions in Display 4.2.

The program should loop until the user enters **0** for length. Use the **pretest while structure** as you did in Project Two. The loop structure should be part of the main function.

The program includes the following functions: Be sure to use the function names provided. Do not change the specifications for the functions.

displayTitle
This function displays a title for the window. *displayTitle* is an example of a **void function** (Section 5.1).

getData
This function displays a prompt requesting the value for the length. If **0** is not entered for the length, the function displays a prompt requesting the width. This function is a **call-by-reference** since it needs to supply main with two values.

Check Display 5.4 getNumbers function and Display 5-9 getInput function.

The rest of the functions are **call-by-value**. Do not use call call-by-reference.

calcSqFt
This function receives the values for length and width, and calculates the square footage including the ceiling.

The function returns an integer to main.

calcTotalCans
This function calculates and returns the number of cans needed rounding up to the next whole number.

calcCost
This function calculates and returns the total cost.

displayResults
This function receives the numeric results to display. Note how the decimals are aligned for readability in the sample run.

displayMsg
This function displays a message reporting a free gift based on the number of cans purchased. Be sure to use a nested if. The customer only receives one free gift.

1 - 3 cans	– free paint brush.
4-7 cans	– free paint tray.
More than 7 cans	– $10 gift card.

Here is a sample run:

Purpose of this project

Develop C++ program with the following new features:

- Nested ifs (pages 120-127)
- Predefined functions (pages 183-188)
- Call-by-value functions (pages 192-228)
- void functions (pages 252-257)
- Call-by-reference (pages 259-271)

Advice from the Instructor:

- Draw a picture to determine what 5 areas you need to calculate for the room.
- Add one function at a time. First code and test *displayTitle*. When it works, code and test *getData*. Continue adding the functions one at a time.
- While testing your code, add temporary couts in main to display the results returned to main from each function.
- The *Average Problem* in the *Chapter Four Notes* contains all the programmer-defined function syntax you need for the call-by-value and void functions.
- The sample programs on pages 193 and 222 are good examples for call-by-value functions.
- See also the *Anatomy of a Function* call in Display 4.4.
- The constants are globals. All variables, however, still must remain **local** variables.
- Be sure that your functions contain only **one** return.
- Be sure to check for redundant code in the displayMsg function.
- Try to correct any warning messages.

CSIT 210 *Introduction to Programming* (Java) is a prerequisite for this course. You will find that C++ statements are very similar to Java so you can rely on your Java skills in this class. Functions and methods are the same concept so you can use your knowledge of Java methods when designing/writing C++ functions.