Composition Demo

1. Create a new C++ Empty Console project in Visual Studio. Name it CompositionDemo.
2. Right-click on the project or on Project in the toolbar and select Add class. Name the class Carpet.
3. Open the Carpet.h file.
4. Add the include guards:

#ifndef CARPET\_H

#define CARPET\_H

#endif

1. Add these private class variables:

Measurement width; // these will be red, don’t panic

Measurement length;

double area;

void CalculateArea();

1. Add these public class methods:

Carpet(int lengthFeet, int lengthInches, int widthFeet, int widthInches);

double GetCarpetArea();

string GetCarpetStr();

1. Let Visual Studio create the implementation stubs for you.
2. Create another class, Measurement. This class will handle the feet and inches measurements.
3. Add these private variables:

int feet; // To hold a number of feet

int inches; // To hold a number of inches

Measurement Simplify(int simpInches);

1. Add these public methods:

Measurement(int f, int i);

// Mutator functions

void SetFeet(const int f){ feet = f; }

void SetInches(const int i){ inches = i; }

// Accessor functions

int GetFeet() const { return feet; }

int GetInches() const { return inches; }

int Multiply(Measurement obj);

double ConvertToDouble(); //Convert feet and inches to decimal feet

1. Let Visual Studio create the implementation stubs for you.
2. Open the measurement.cpp file.
   1. Add #include <cmath>
   2. Complete the Constructor initializer:

Measurement::Measurement(int f, int i) : feet{ f }, inches{i} {}

1. Complete the Simplify method:

int simpFeet{0};

if (simpInches >= 12)

{

simpFeet += (simpInches / 12);

simpInches = simpInches % 12;

}

else if (simpInches < 0)

{

simpFeet -= ((abs(simpInches) / 12) + 1);

simpInches = 12 - (abs(simpInches) % 12);

}

return Measurement(simpFeet, simpInches);

1. Complete the Multiply method:

int inches1 = feet \* 12 + inches;

int inches2 = obj.feet \* 12 + obj.inches;

int simpInches = inches1 \* inches2;

return simpInches;

1. Complete the ConvertTodouble method:

double temp = feet;

temp += (inches / 12.0);

return temp;

1. Now go to the Carpet class and #include “Measurement.h”
2. Now we can complete the Carpet.cpp file.
3. First, the constructor. The arguments to the Carpet constructor are the length and width of a rectangular carpet in feet and inches. Those measurements do not exist in the Carpet class. We only have the Measurement object length and width. So we need to call the constructor of the Measurement class as well:

Carpet:: Carpet(int lengthFeet, int lengthInch, int widthFeet, int widthInch) :

length(Measurement(lengthFeet, lengthInch)),width(Measurement(widthFeet, widthInch))

Then, call CalculateArea();

1. Now CalculateArea()
   1. Before we start, add a constant to the Carpet.cpp file:

const double SQ\_IN\_TO\_YARDS = 1296.0;

* 1. Now we will use our Measurement objects, length and width:

area = (length.Multiply(width))/ SQ\_IN\_TO\_YARDS;

* 1. So the answer is in square yards

1. Inline the method GetCarpet area in the Carpet.h file:
   1. double GetCarpetArea() { return area; }
   2. Be sure to delete any code in the carpet.cpp file – it will be a duplicate.
2. And the last method in the carpet class is GetCarpetStr(). I like to use stringstream, so add an import of <sstream> to the file and:

stringstream ss;

ss <<"\n The carpet is "<< length.GetFeet() << " feet, "

<< length.GetInches() << " inches" << " by "<< width.GetFeet() << " feet, "

<< width.GetInches() << " inches"

<<"\n The area of the carpet is " << area << " square yards";

return ss.str();

1. Create a .cpp file and call it Driver.cpp.
   1. Write your header
   2. #include <iostream. And #include “carpet”
   3. Create int main()
   4. cout your intro:

cout << "\n Composition Demo - the Area of a rectangular carpet.";

* 1. Declare some local variables:

int lengthFt{ 1 };

int lengthIn{ 1 };

int widthFt{ 1 };

int widthIn{ 1 };

* 1. Ask the user for the length and width of their carpet, in feet and inches.

cout << "\n Please enter the length of your carpet in feet, then a space , and then inches."

<< "\n please use whole numbers: ";

cin >> lengthFt >> lengthIn;

cout << "\n Please enter the width of your carpet in feet, then a space , and then inches."

<< "\n please use whole numbers: ";

cin >> widthFt >> widthIn;

* 1. Create a carpet object using its overloaded constructor:

Carpet carpet{ lengthFt, lengthIn, widthFt, widthIn };

* 1. Display the results:

cout << carpet.GetCarpetStr();

* 1. Say thank you;

cout << "\n Thanks for doing our composition demo";

1. Run your project. Calculate some carpet areas.