```
 1  /* Filename: main.cpp
 2   * ------------------
 3   * Class:   CSCI 162
 4   * Lab:     #18 (part 2)
 5   * Author:  Darwin Jacob Groskleg
 6   * Date:    March 1, 2018
 7   *
 8   * Purpose: to draw a rectangle of a given dimensions to the screen (cout)
 9   * while using classes to implement it. Also implement a private method that
10   * is called by the draw() method (see Rectangle#drawTopOrBottom).
11   *
12   * Sample Output:
13   * ------------------------------------------------------------------------
14   * Enter rectangle width:  8.8
15   * Enter rectangle height: 3.45
16   *
17   * Rendering with 9x3 pixels:
18   *  _________
19   * |         |
20   * |         |
21   * |         |
22   *  _________
23   * A 8.8x3.45 rectangle.
24   *
25   * #> ./rect
26   * Enter rectangle width:  5
27   * Enter rectangle height: 0.3
28   *
29   * Rendering with 5x0 pixels:
30   *  _____
31   * A 5x0.3 rectangle.
32   * ------------------------------------------------------------------------
33   */
34  #include <iostream>
35  #include "rectang.h"
36
37  using namespace std;
38
39  int main() {
40      double width = 0.0;
41      double height = 0.0;
42
43      cout << "Enter rectangle width:  ";
44      cin >> width;
45
46      cout << "Enter rectangle height: ";
47      cin >> height;
48
49      Rectangle r(width, height);
50      r.draw();
51      r.describe();
52
53      return 0;
54  }
```

```
 1  /* Filename: rectang.cpp
 2   * ---------------------
 3   * Class:   CSCI 162
 4   * Lab:     #18 (part 2)
 5   * Author:  Darwin Jacob Groskleg and Martin van Bommel
 6   * Date:    March 1, 2018
 7   *
 8   * Purpose: interface for rectangle objects. So far it is not appropriate for
 9   * our getters and setters to be public.
10   */
11  #ifndef RECTANG_H_INCLUDED
12  #define RECTANG_H_INCLUDED
13
14  class Rectangle
15  {
16      private:
17          double width;
18          double height;
19          void   setWidth  (double);
20          void   setHeight (double);
21          double getWidth  () const { return width; }
22          double getHeight () const { return height; }
23          void   drawTopOrBottom(int) const;
24      public:
25          // Constructor
26          Rectangle(double w, double h) {
27              setWidth(w);
28              setHeight(h);
29          }
30          void   describe  () const;
31          void   draw      () const;
32          double getArea   () const { return height * width; }
33  };
34
35  #endif // RECTANG_H_INCLUDED
```

```
 1  /* Filename: rectang.h
 2   * --------------------
 3   * Class:   CSCI 162
 4   * Lab:     #18 (part 2)
 5   * Author:  Darwin Jacob Groskleg and Martin van Bommel
 6   * Date:    March 1, 2018
 7   *
 8   * Purpose: implements methods on Rectangle object.
 9   */
10  #include "rectang.h"
11  #include <iostream>
12  #include <cmath>
13
14  using namespace std;
15
16  /* Prints a description of the rectangle, its dimensions.
17   */
18  void Rectangle::describe() const {
19      cout << "A " << width << "x" << height << " rectangle.\n" << endl;
20  }
21
22  /* Sets the width while checking for valid input.
23   */
24  void Rectangle::setWidth(double w) {
25      if (w < 0) {
26          cout << "INVALID INPUT: Cannot have negative width (given " << w
27               << ") of rectangle!" << endl;
28          exit(0);
29      }
30      else width = w;
31  }
32
33  /* Sets the height while checking for valid input.
34   */
35  void Rectangle::setHeight(double h) {
36      if (h < 0) {
37          cout << "INVALID INPUT: Cannot have negative height (given " << h
38               << ") of rectangle!" << endl;
39          exit(0);
40      }
41      else height = h;
42  }
43
44  /* Method: draw
45   * Usage: rectange.draw();
46   * -----------------------
47   * Prints to characters to stdout that represent the rectangle with its proper
48   * dimensions so that it can be displayed on the console screen. In order to d
49   * this the dimensions are rounded to the nearest integer value. Note that eve
50   * an empty rectangle (0x0) will take up 2 empty lines of space, or 2 rows and
51   * 2 columns minimum.
52   */
53  void Rectangle::draw() const {
54      int w = (int) nearbyint(width);
55      int h = (int) nearbyint(height);
56
57      cout << "\nRendering with " << w << 'x' << h << " pixels:"<< endl;
58      drawTopOrBottom(w);
59      for (int row=1; row<h+1; row++) {
60          for (int col=0; col<w+2; col++) {
61              if (col==0 || col==w+1)
62                  cout << '|';        // Side
63              else
64                  cout << ' ';        // Inside
65          }
66          cout << endl;
67      }
68      // The vertical gap is too big, just draw a line
69      if (h != 0) drawTopOrBottom(w);
```

```
70  }
71
72  /* Method: drawTopOrBottom
73   * -----------------------
74   * Pivate method used to print the top and bottom edges of the rectangle in
75   * ascii to stdout.
76   */
77  void Rectangle::drawTopOrBottom(int draw_width) const {
78      cout << ' ';                    // Corners
79      for (int col=1; col<draw_width+1; col++)
80          cout << '-';                // Top & Bottom Edges
81      cout << ' ' << endl;            // Corners
82  }
```