CISC2000: Computer Science II LAB

**rectangle class**

rectangle.cpp

Below is some code for a rectangle class that needs to be completed. The goal is to code the class so that it works without changing the main program.

# Starter Code

This is the starter code. Declare the member functions, deﬁne the accessors and ﬁll in the member function code.

#include <iostream> using namespace std;

// rectangle has a vertical height and horizontal width

// The class below is a rectangle. It has two private

// data members: height and width.

// TODO: Complete the class declaration and definition. class rectangle {

public:

// TODO: declare a default constructor that sets height & width to 1.

// TODO: declare member function void add

// @param int height, int width

// TODO: declare member function void set

// @param int height, int width

// TODO: declare member function void draw

// TODO: define accessor for width (remember it must return int)

// TODO: define accessor for height

// TODO: define a function to tell if a rectangle is a square private:

int width, height;

};

// TODO: Implement a default constructor to initialize height and width to 1.

// The default constructor executes when the object is first declared.  
// It is a function with the same name as the class (see below).

// It also has no return type

rectangle::rectangle()

{

}

// TODO: Implement add to increment the dimensions

// add the given addHeight to height and addWidth to width

void rectangle::add(int addHeight, int addWidth)

{

}

// TODO: Implement set to overwrite the data members void rectangle::set(int h, int w)

{

}

// TODO: Implement draw to draw a rectangle using '#' characters void rectangle::draw()

{

}

// TODO: Define getWidth and getHeight

// TODO: Implement isSquare to indicate if a rectangle is a square. int main()

{

// Declare 2 rectangles rectangle r1, r2;

// Print unit rectangle

cout << "unit rectangle" << endl; r1.draw();

// Set, print dimensions and draw r1.set(4, 3);

cout << "r1 is " << r1.getHeight() << " x " << r1.getWidth() << endl;

r1.draw();

// Assign, increment, print dimensions and draw r2 = r1;

r2.add(3, 4);

cout << "r2 is " << r2.getHeight() << " x " << r2.getWidth() << endl;

r2.draw();

if (r2.isSquare())

cout << "r2 is a square" << endl; else

cout << "r2 is not a square" << endl;

return 0;

}

# Program Output

unit rectangle #

r1 is 4 x 3 ###

### ### ###

r2 is 7 x 7 ####### ####### ####### ####### ####### ####### #######

r2 is a square