```
/*************************************************************
 *
 * Project 2: Extending the GUI
 *
 * Author: Nicholas Primiano <nprimiano@fordham.edu>
 * Date: 9 October 2013
 *
 * Extend the Shape class adding a class Square.
 *

 *************************************************************/
#include "Square.h"

int main()
    try{
        using namespace Graph_lib;

        Point window_point(100,100);  //top left point of window
        Point length_text (20,20);    // top left point of line 1 text
        Point next_text (20, 50);     //top elft point of next line of text

        //define a simple window
        Simple_window square_window(window_point,600,600,"Square");

        int length_0  =80;  //side length of square2
        int length_1 = 125;  //side length of square4

        //define proper squares
        Square square1(Point(150,150),Point(250,250));
        Square square2(Point(300,200),length_0);

        Square square3(Point(200,200),Point(300,300));
        Square square4(Point(350,350),length_1);

        //declare and define ostreams for descriptive text
        ostringstream oss_square;
        ostringstream oss_square_error;
        ostringstream oss_next_1;
        ostringstream oss_next_2;

        oss_square << "The side-length of the blue-outlined square is " << length_0 <<" pixels";
        oss_next_1 << "Press the ""Next"" button to change the picture";

        oss_square_error << "The side-length of the white square is " << length_1 << " pixels.";
        oss_next_2 << "Press the ""Next"" button for an unpleasant suprise";

        Text square_window_square (length_text, oss_square.str());
        Text square_window_next_1 (next_text, oss_next_1.str());

        //style text
        square_window_square.set_font_size(18);
        square_window_next_1.set_font_size(18);

        Text square_window_error_text (length_text, oss_square_error.str());
        Text square_window_next_2_text (next_text, oss_next_2.str());

        //style text
        square_window_error_text.set_font_size(18);
        square_window_next_2_text.set_font_size(18);

        //stly proper squares
        square1.set_color(Color::blue);
        square2.set_color(Color::red);
        square3.set_color(Color::red);
        square4.set_fill_color(Color::white);
        square4.set_style(Line_style(Line_style::dash,2));

        square_window.set_label("Project 2");
```



```
        //attach first elemetnts, proprer Squares and text
        square_window.attach(square_window_square);
        square_window.attach(square_window_next_1);
        square_window.attach(square1);
        square_window.attach(square2);
        square_window.wait_for_button();

        //remove first elements
        square_window.detach(square_window_square);
        square_window.detach(square_window_next_1);
        square_window.detach(square1);
        square_window.detach(square2);

        //attach second elements, Squares and test
        square_window.attach(square_window_error_text);
        square_window.attach(square_window_next_2_text);
        square_window.attach(square3);
        square_window.attach(square4);
        square_window.wait_for_button();

        //error : not a Square
        Square square5(Point(200,200),Point(300,299));

        square_window.attach(square5);
        square_window.wait_for_button();

    }
    catch(int e){
        if (e == 1)
            cerr << ("Bad Square: non-positive side length");
        if (e == 2)
            cerr << ("Bad Square: side lengths not equal\n");
    }
```

```
#include "Simple_window.h"
#include "Graph.h"
#include <iostream>



class Square : public Shape {

public:


//constructor for Square defined by single point and side length
    Square(Point xy, int ll) : l(ll)
        {
            if (l <= 0){
                throw 1;
            }
            add(xy);
        }
//constructor for Square defined by two points
    Square(Point point1, Point point2) : l(point2.x-point1.x)
        {

            if(l != (point2.y-point1.y)){
                throw 2;
            }
            if(l <=  0){
                throw 1;
            }

            add(point1);
        }
//draw square
    void draw_lines() const;

    int side() const { return l;}

private:
    int l;    //side length
};
```



```
#include "Square.h"

//draw square
void Square :: draw_lines() const{


    if (fill_color().visibility()) {    // fill
        fl_color(fill_color().as_int());
        fl_rectf(point(0).x,point(0).y,l,l);
    }

    if (color().visibility()) {    // lines on top of fill
        fl_color(color().as_int());
        fl_rect(point(0).x,point(0).y,l,l);
    }
}
```