Reference Manual

Generated by Doxygen 1.8.13

Contents

1 Test List

Member TEST_CASE ("contains")

contains() member function to test Tests rectangle r1 10 by 10 contains points (5,5), (10,10),(11,13)

```
labCh10a_2.cpp:29: passed: ss.str() =="truetruefalse" for: "truetruefalse" == "truetruefalse"
Passed 1 test case with 1 assertion.
```

contains() member function to test Tests rectangle r2 5 by 5 contains points (-1,5),(5,9),(5,3)

```
labCh10a_2.cpp:44: passed: ss.str() =="falsefalsetrue" for: "falsefalsetrue" == "falsefalsetrue"
Passed 1 test case with 1 assertion.
```

contains() member function to test Tests rectangle r3 1 by 3 contains points (5,5), (11,10),(11,13)

```
labCh10a_2.cpp:57: passed: ss.str() =="falsetruetrue" for: "falsetruetrue" == "falsetruetrue"
Passed 1 test case with 1 assertion.
```

2 Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

3 File Index

3.1 File List

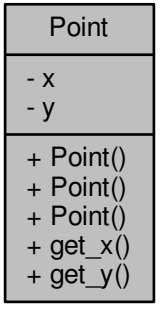
Here is a list of all files with brief descriptions:

4 Class Documentation

4.1 Point Class Reference

`#include <Point.h>`

Collaboration diagram for Point:



Public Member Functions

- Point ()
- Point (double _x, double _y)
- Point (const Point &p)
- double get_x () const
- double get_y () const

Private Attributes

- double x
- double y

4.1.1 Detailed Description

A Point on a (x,y) coordinate with a x and y coordinate.

Definition at line 7 of file Point.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Point() [1/3]

```
Point::Point ( )
```

The default constructor sets Point to (0.0,0.0)

Definition at line 3 of file Point.cpp.

```
4       :x(0.0),y(0.0)
5 {}
```

4.1.2.2 Point() [2/3]

```
Point::Point (
            double _x,
            double _y )
```

A two-argument constructor sets the Point to its arguments

Parameters

↩ _↩ *x*	the x coordiante
↩ _↩ *y*	the y coordiante

Definition at line 6 of file Point.cpp.

```
7       :x(_x),y(_y)
8 {}
```

4.1.2.3 Point() [3/3]

```
Point::Point (
            const Point & p )
```

The copy constructor sets the Point to its argument

Parameters

p	the reference to another Point p

Definition at line 9 of file Point.cpp.

```
10      :x(p.x),y(p.y)
11 {}
```

4.1.3 Member Function Documentation

4.1.3.1 get_x()

```
double Point::get_x ( ) const
```

Returns

x value

Definition at line 12 of file Point.cpp.

```
13 {
14         return x;
15 }
```

Here is the caller graph for this function:

4.1.3.2 get_y()

```
double Point::get_y ( ) const
```

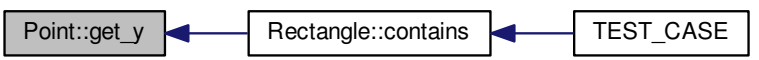
Returns

y value

Definition at line 16 of file Point.cpp.

```
17 {
18         return y;
19 }
```

Here is the caller graph for this function:



4.1.4 Member Data Documentation

4.1.4.1 x

```
double Point::x  [private]
```

Definition at line 34 of file Point.h.

4.1.4.2 y

```
double Point::y  [private]
```

Definition at line 34 of file Point.h.

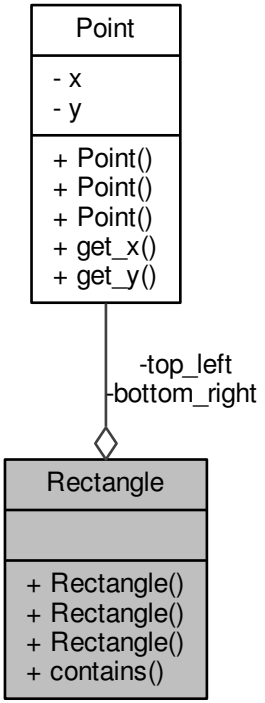
The documentation for this class was generated from the following files:

- Point.h
- Point.cpp

4.2 Rectangle Class Reference

```
#include <Rectangle.h>
```

Collaboration diagram for Rectangle:



Public Member Functions

- Rectangle ()
- Rectangle (Point &tl, Point &br)
- Rectangle (const Rectangle &p)
- bool contains (Point &p) const

Private Attributes

- Point top_left
- Point bottom_right

4.2.1 Detailed Description

A Rectangle on a (x,y) coordinate Defined as two points with vertical lines and horizontal lines to form the rectangle.

Definition at line 8 of file Rectangle.h.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Rectangle() [1/3]

```
Rectangle::Rectangle ( )
```

The default constructor sets Rectangle to have two points with (0.0,0.0)

Definition at line 3 of file Rectangle.cpp.

```
4       :top_left(),bottom_right()
5 {}
```

4.2.2.2 Rectangle() [2/3]

```
Rectangle::Rectangle (
            Point & tl,
            Point & br )
```

A two-argument constructor sets the Rectangle to its arguments also checks which point is on tl or br

Parameters

tl	the top left point (or any point)
br	the bottom right point (or any point)

Definition at line 6 of file Rectangle.cpp.

```
7       :top_left (((tl.get_x() <= br.get_x())? tl.get_x() : br.
        get_x()),
8       ((tl.get_y() <= br.get_y()) ? tl.get_y() : br.get_y())),
9       bottom_right(((tl.get_x() >= br.get_x())? tl.get_x() : br.
        get_x()),
10       ((tl.get_y() >= br.get_y()) ? tl.get_y() : br.get_y()))
11  {}
```

4.2.2.3 Rectangle() [3/3]

```
Rectangle::Rectangle (
            const Rectangle & p )
```

The copy constructor sets the Rectangle to its argument

Parameters

p	the reference to another Rectangle p

Definition at line 12 of file Rectangle.cpp.

```
13       :top_left(p.top_left),bottom_right(p.
        bottom_right)
14  {}
```

4.2.3 Member Function Documentation

4.2.3.1 contains()

```
bool Rectangle::contains (
            Point & p ) const
```

This will check if the Rectangle contains a Point

Parameters

p	the point being checked

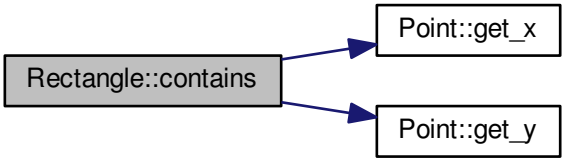
Returns

bool value if the Rectangle contains a point true for if Rectangle contains Point false if Rectangle does not contain Point

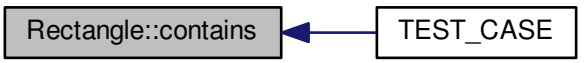
Definition at line 15 of file Rectangle.cpp.

```
16 {
17     if (p.get_x() >= top_left.get_x() &&
18         p.get_x() <= bottom_right.get_x() &&
19         p.get_y() >= top_left.get_y() &&
20         p.get_y() <= bottom_right.get_y())
21         return true;
22     return false;
23 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.4 Member Data Documentation

4.2.4.1 bottom_right

```
Point Rectangle::bottom_right  [private]
```

Definition at line 39 of file Rectangle.h.

4.2.4.2 top_left

```
Point Rectangle::top_left  [private]
```

Definition at line 38 of file Rectangle.h.

The documentation for this class was generated from the following files:

- Rectangle.h
- Rectangle.cpp

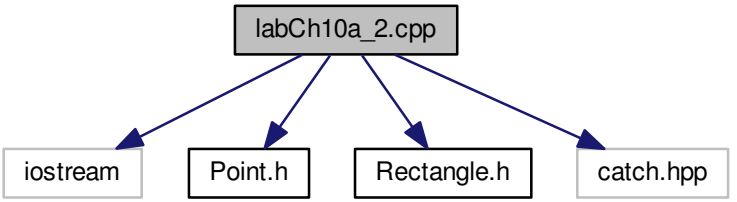
5 File Documentation

5.1 labCh10a_2.cpp File Reference

```
#include <iostream>
#include "Point.h"
#include "Rectangle.h"
```

```
#include "catch.hpp"
```

Include dependency graph for labCh10a_2.cpp:



Functions

- TEST_CASE ("contains")

5.1.1 Function Documentation

5.1.1.1 TEST_CASE()

```
TEST_CASE (
            "contains"  )
```

Test contains() member function to test Tests rectangle r1 10 by 10 contains points (5,5), (10,10),(11,13)

```
labCh10a_2.cpp:29: passed: ss.str() =="truetruefalse" for: "truetruefalse" == "truetruefalse"
Passed 1 test case with 1 assertion.
```

Test contains() member function to test Tests rectangle r2 5 by 5 contains points (-1,5),(5,9),(5,3)

```
labCh10a_2.cpp:44: passed: ss.str() =="falsefalsetrue" for: "falsefalsetrue" == "falsefalsetrue"
Passed 1 test case with 1 assertion.
```

Test contains() member function to test Tests rectangle r3 1 by 3 contains points (5,5), (11,10),(11,13)

```
labCh10a_2.cpp:57: passed: ss.str() =="falsetruetrue" for: "falsetruetrue" == "falsetruetrue"
Passed 1 test case with 1 assertion.
```

Definition at line 6 of file labCh10a_2.cpp.

```
7 {
8     Point p1(10,10);
9     Point p2(0,0);
10     Point p3(5,5);
11     Point p4(11,13);
```

```
12      Rectangle r1(p1,p2);
13      Rectangle r2(p3,p2);
14      Rectangle r3(p4,p1);
15      std::streambuf *b = std::cout.rdbuf();
16      std::stringstream ss;
17      std::streambuf *sb = ss.rdbuf();
18      std::cout.rdbuf(sb);
24      SECTION("r1")
25      {
26          (r1.contains(p3))? std::cout<<"true" : std::cout<<"false";
27          (r1.contains(p1))? std::cout<<"true" : std::cout<<"false";
28          (r1.contains(p4))? std::cout<<"true" : std::cout<<"false";
29          CHECK(ss.str() =="truetruefalse");
30      }
36      SECTION("r2")
37      {
38          p3 = Point (-1,5);
39          (r2.contains(p3))? std::cout<<"true" : std::cout<<"false";
40          p1 = Point (5,9);
41          (r2.contains(p1))? std::cout<<"true" : std::cout<<"false";
42          p4 = Point (5,3);
43          (r2.contains(p4))? std::cout<<"true" : std::cout<<"false";
44          CHECK(ss.str() =="falsefalsetrue");
45      }
51      SECTION("r3")
52      {
53          (r3.contains(p3))? std::cout<<"true" : std::cout<<"false";
54          p1 = Point(11,10);
55          (r3.contains(p1))? std::cout<<"true" : std::cout<<"false";
56          (r3.contains(p4))? std::cout<<"true" : std::cout<<"false";
57          CHECK(ss.str() =="falsetruetrue");
```

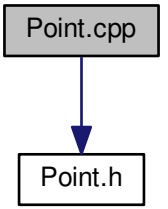
```
58      }
59      std::cout.rdbuf(b);
60  }
```

Here is the call graph for this function:

5.2 Point.cpp File Reference

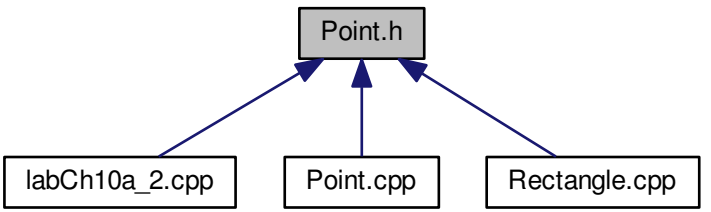
```
#include "Point.h"
```

Include dependency graph for Point.cpp:



5.3 Point.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Point

Macros

- #define POINT

5.3.1 Macro Definition Documentation

5.3.1.1 POINT

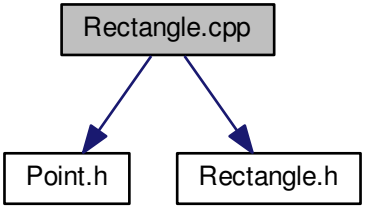
```
#define POINT
```

Definition at line 2 of file Point.h.

5.4 Rectangle.cpp File Reference

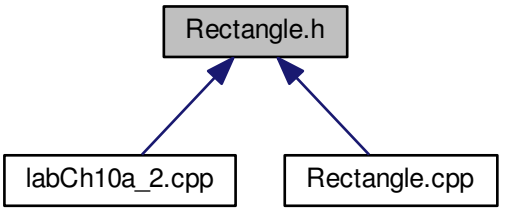
```
#include "Point.h"
#include "Rectangle.h"
```

Include dependency graph for Rectangle.cpp:



5.5 Rectangle.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class Rectangle

Macros

- #define RECTANGLE

5.5.1 Macro Definition Documentation

5.5.1.1 RECTANGLE

```
#define RECTANGLE
```

Definition at line 2 of file Rectangle.h.

Index