Com S 417
Software Testing

Fall 2017 – Week 6, Lecture 9

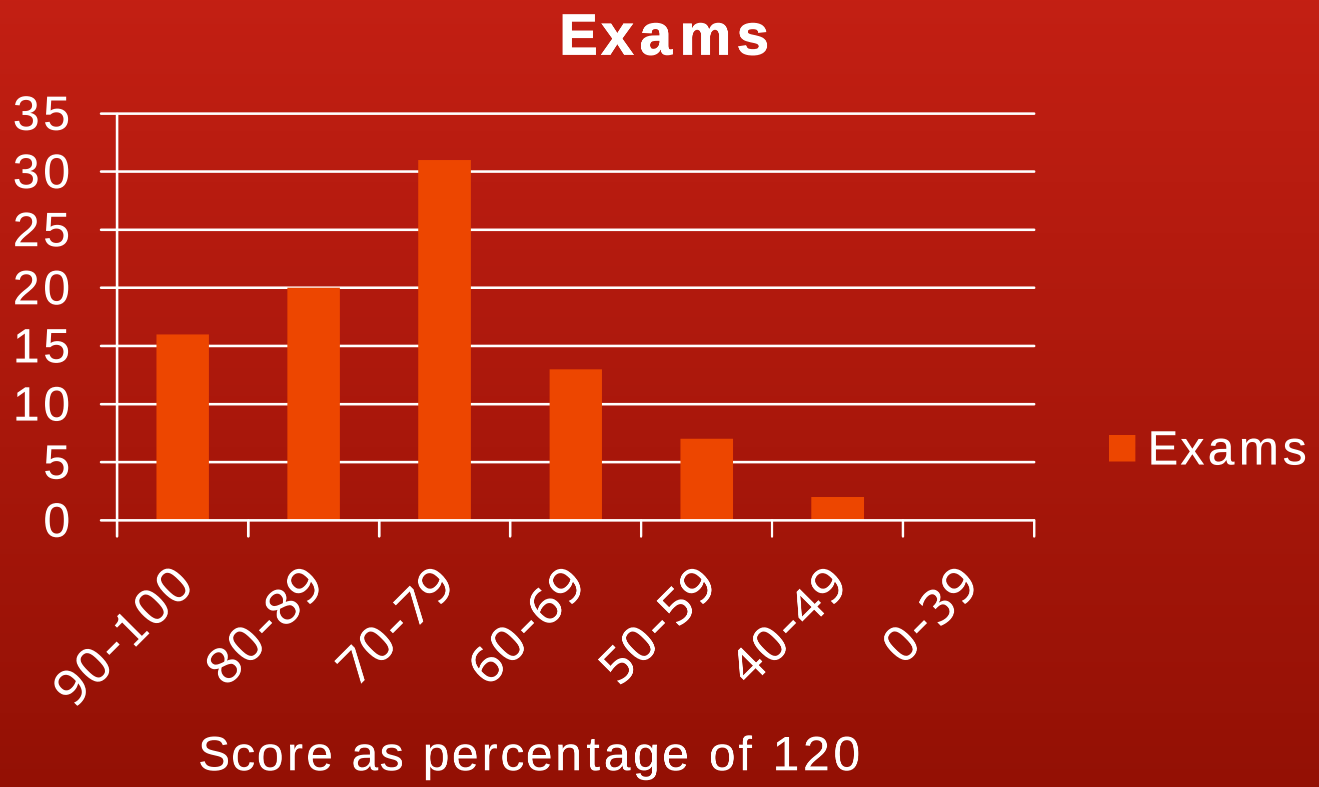
Announcements

- No Office Hours Tomorrow for Robert

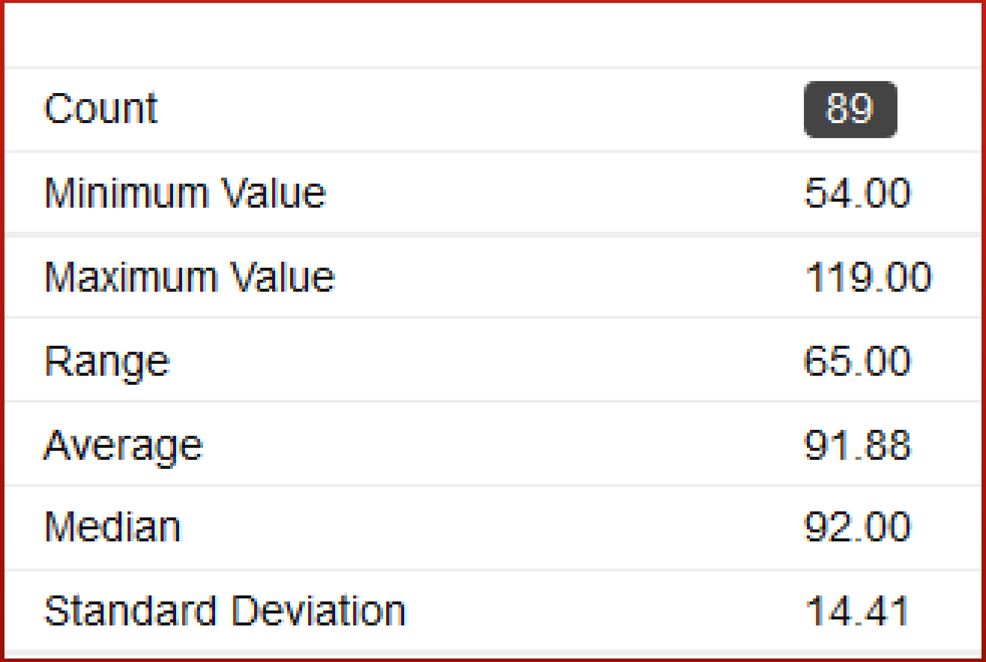
Topics

- Exam Results
- Test Sets in Context
- Computing Test Set Size
- Predicate Coverage Criteria
- Satisfaction
  - Finding input values that satisfy the pre-requisites for a given path or predicate criteria.

Exam Results



Exam Results - Stats



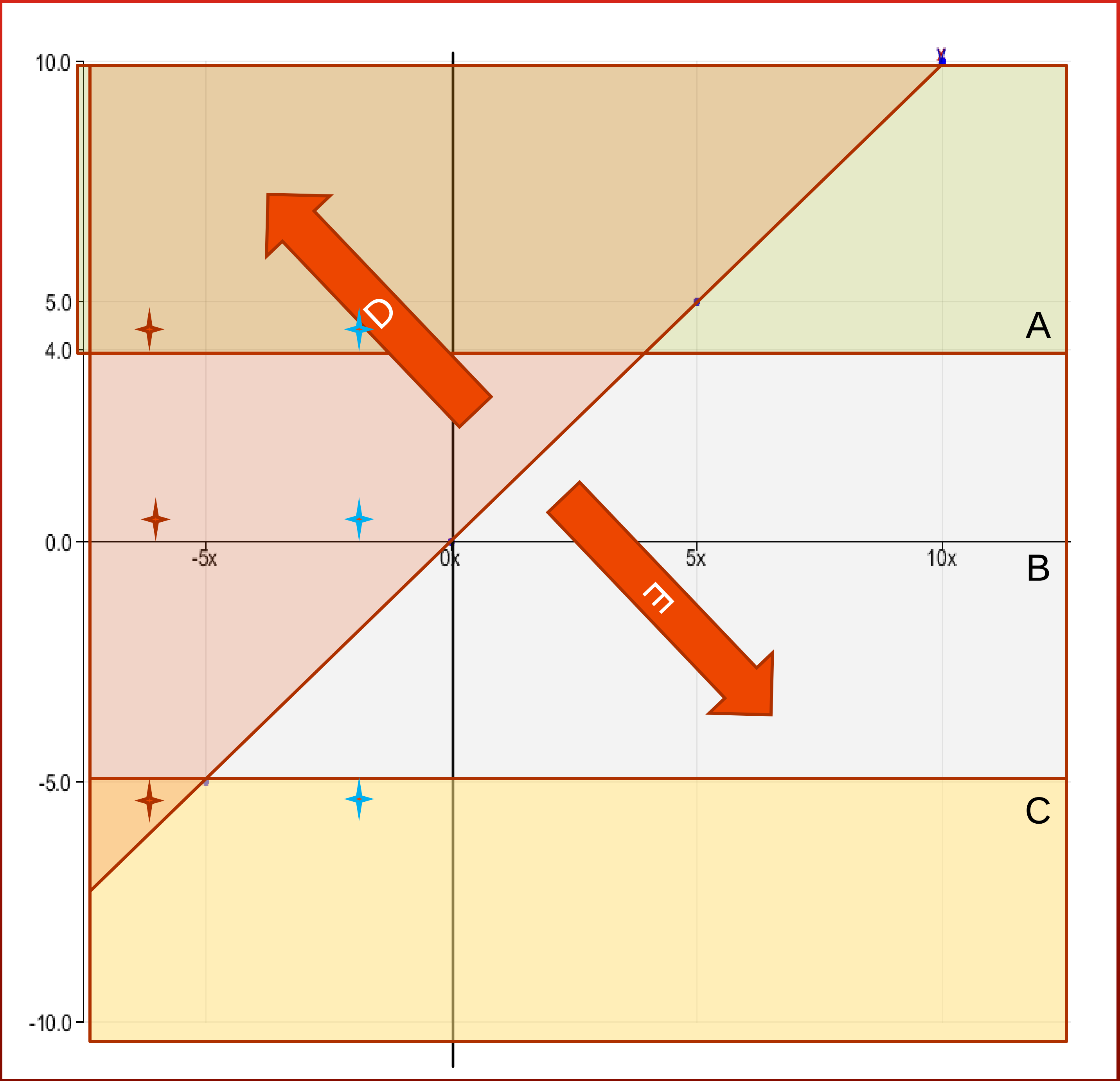
Count	89
Minimum Value	54.00
Maximum Value	119.00
Range	65.00
Average	91.88
Median	92.00
Standard Deviation	14.41

All Combinations

What is the goal?

Why All Combinations and not just one per partition?

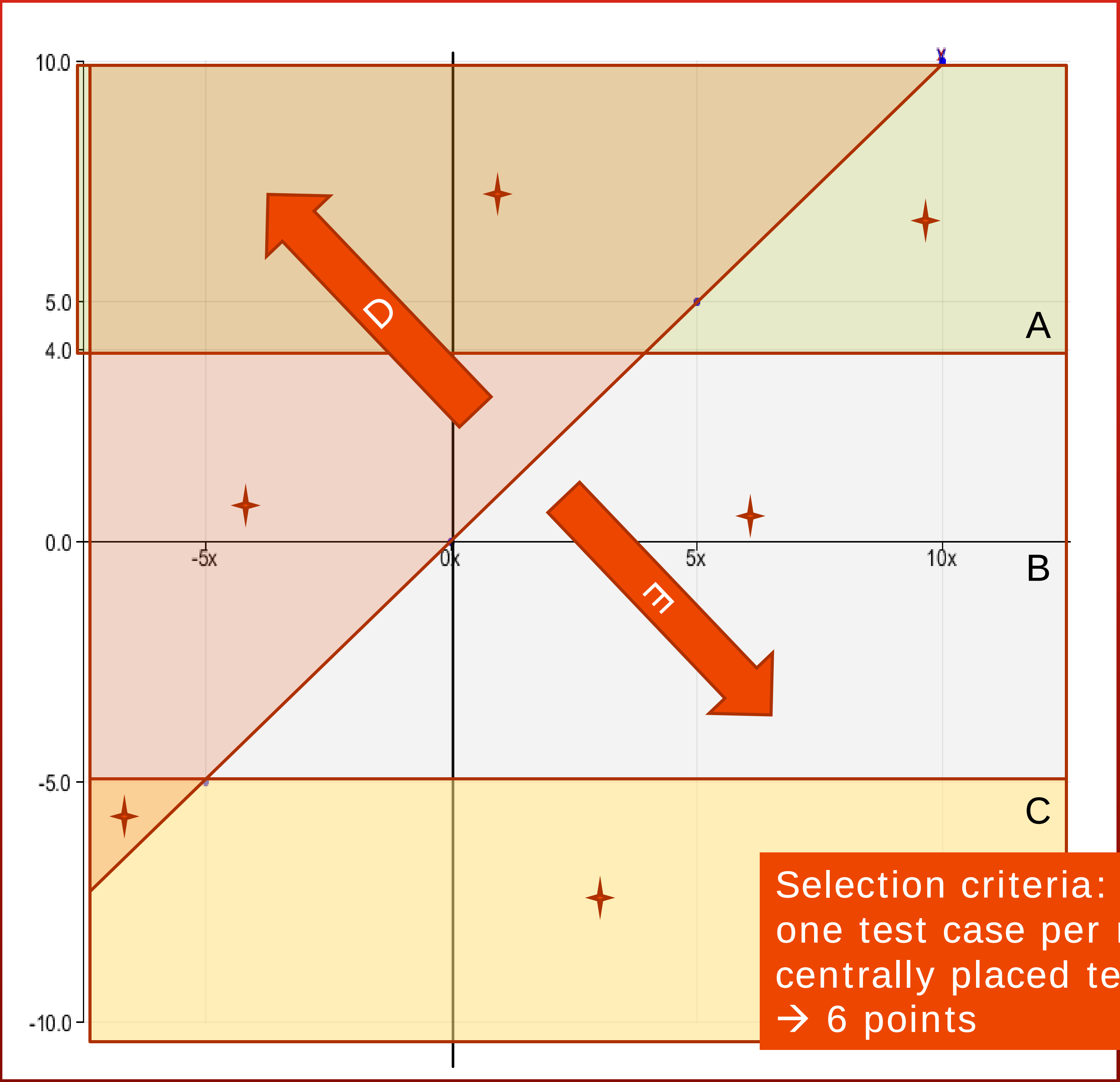
2.11 An application takes two inputs x and y where $x \leq y$ and $-5 \leq y \leq 4$. (a) Partition the input domain using unidimensional partitioning. Derive test sets based on the partitions created in (a).



A: $y>4$
B: $-5..4$
C: $y< -5$

D : $x <= y$
E: $x > y$

What is wrong?



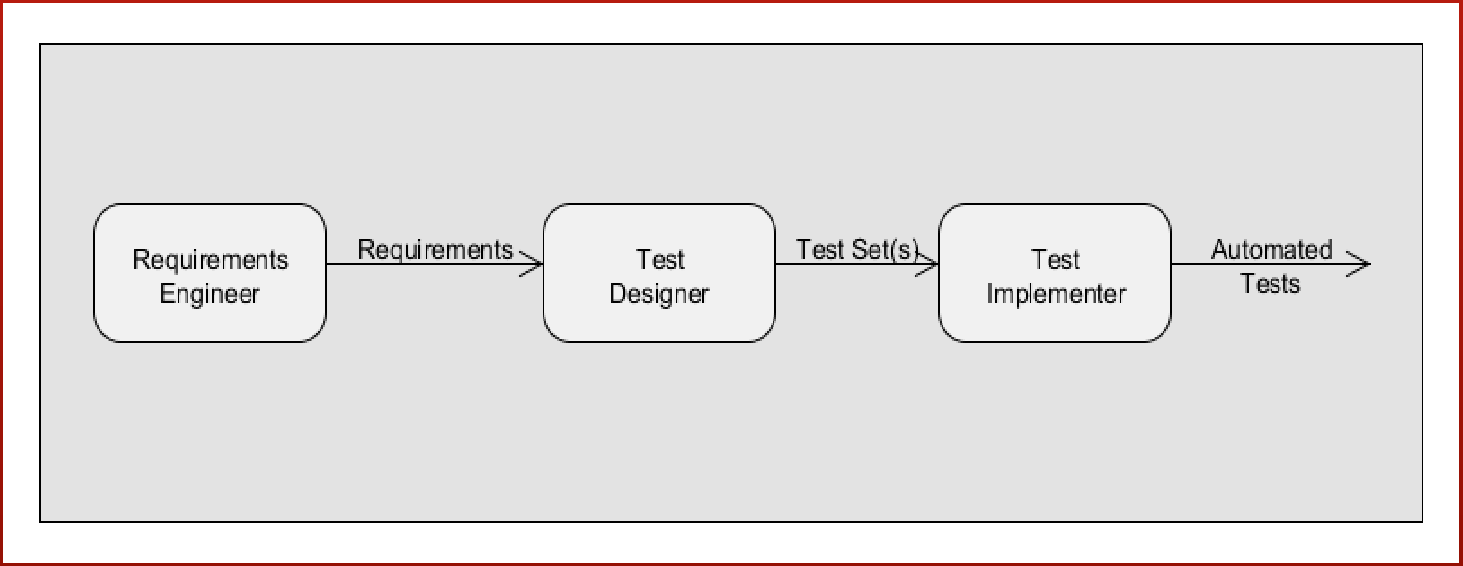
A: y>4
B: -5..4
C: y< -5

D : x <= y
E: x > y

Selection criteria:
one test case per region,
centrally placed tests
→ 6 points

Who Uses a Test Design?

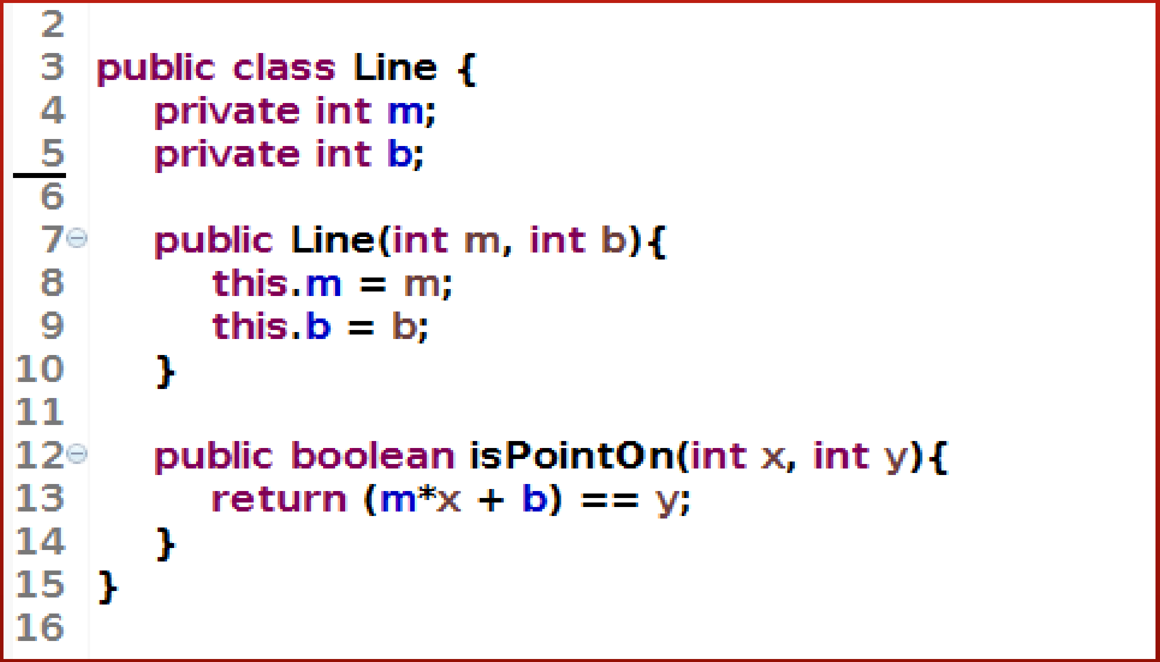
- If there is maximum separation of roles, it would look like this:



- What does the implementer need from the designer?

An Example

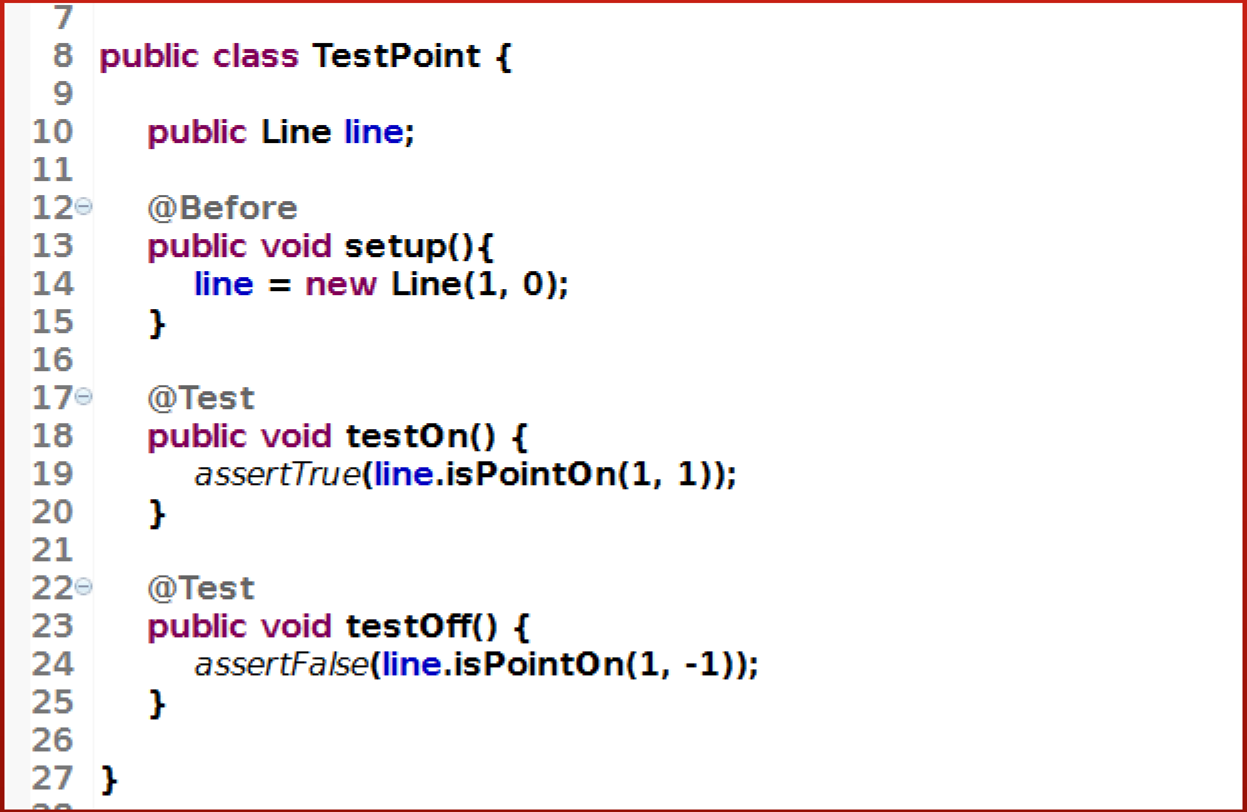
- Like many classes, this class has initial state.
- The method takes inputs and returns a result.



```java

public class Line {
    private int m;
    private int b;

    public Line(int m, int b){
        this.m = m;
        this.b = b;
    }

    public boolean isPointOn(int x, int y){
        return (m*x + b) == y;
    }
}

```

- What do you need to know to write the test?

An Example – the test



```

public class TestPoint {

    public Line line;

    @Before
    public void setup(){
        line = new Line(1, 0);
    }

    @Test
    public void testOn() {
        assertTrue(line.isPointOn(1, 1));
    }

    @Test
    public void testOff() {
        assertFalse(line.isPointOn(1, -1));
    }

}
```

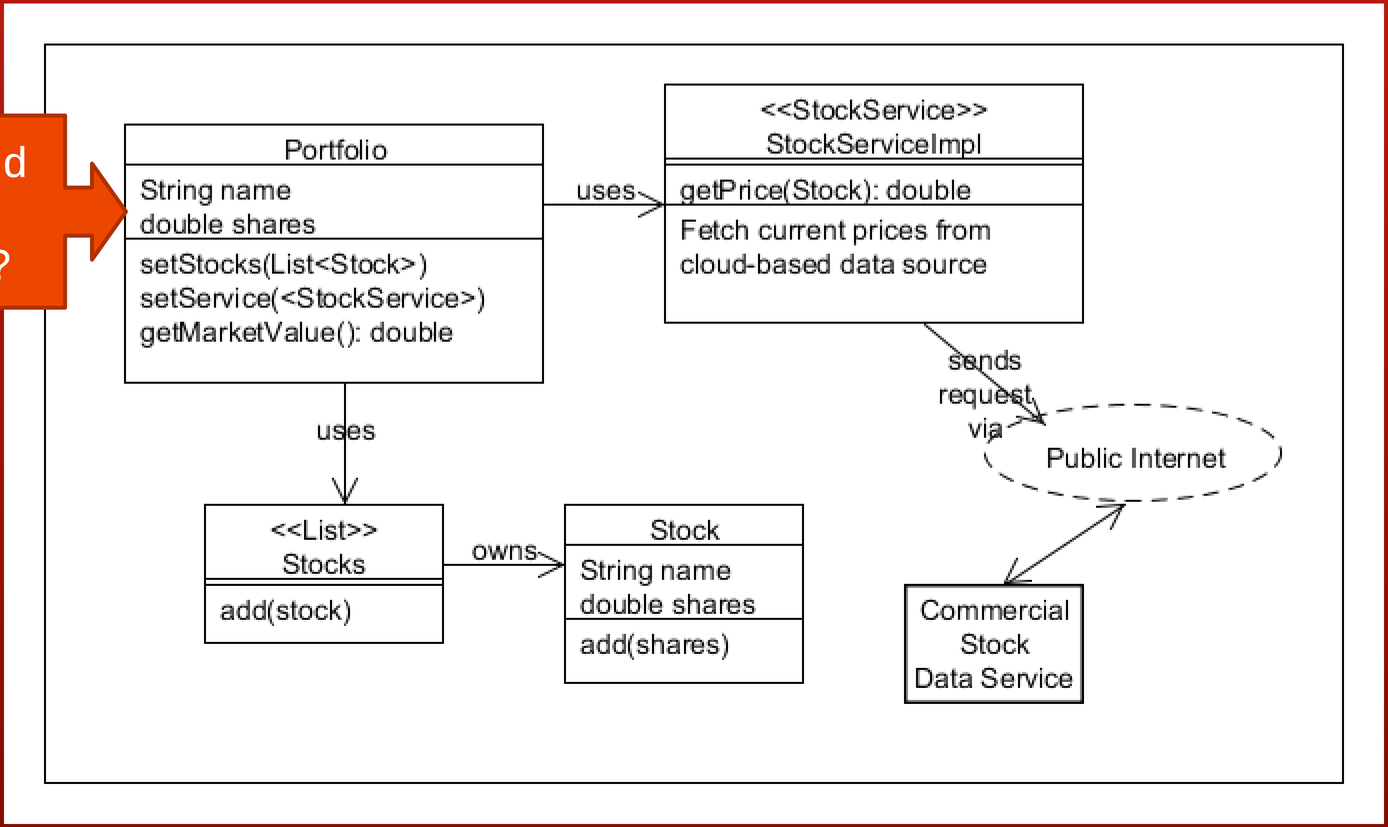
A complete design for one Test Case requires:
<initial conditions, input values, expected behavior>

Sizing Test Sets

Isolating Components for Test

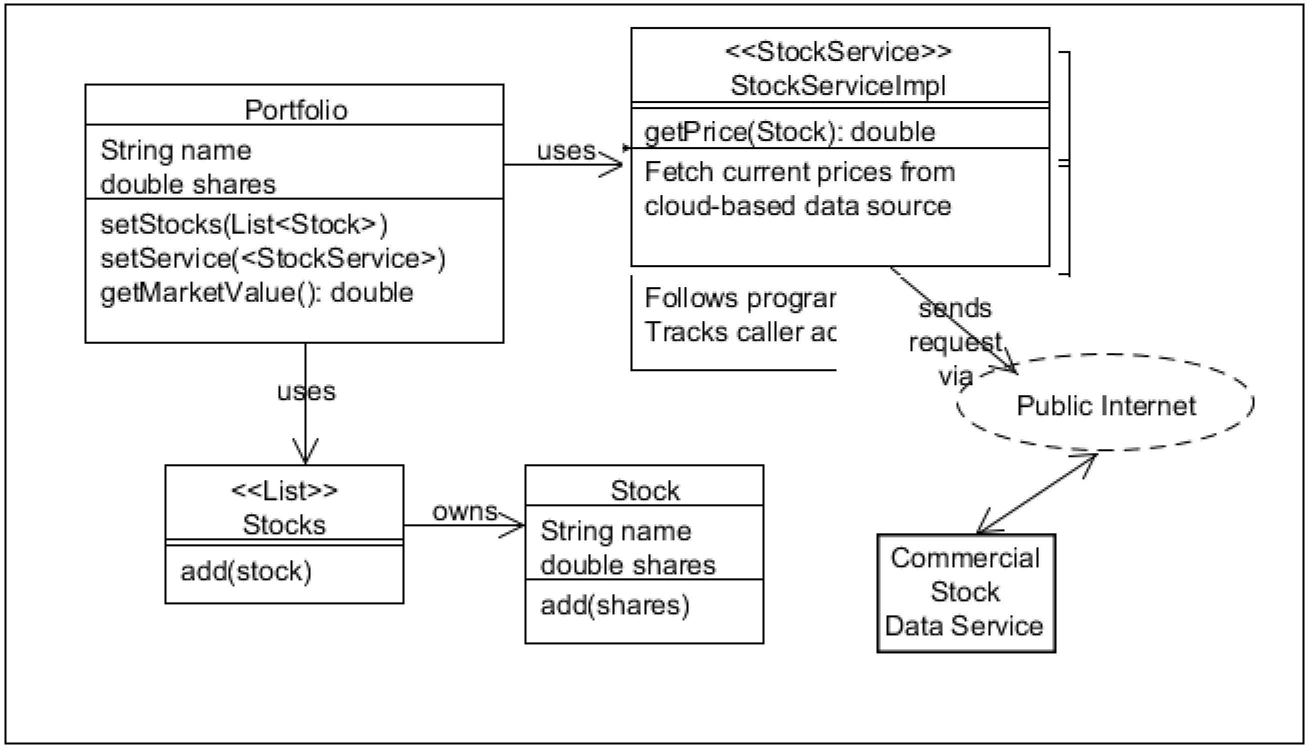
So far we've tested only single methods. The real world is more complex.

How would you test Portfolio?



Oh my ....

Stubs and Mocks and Proxies



What design choices make this test possible?

Reading Assignment

- https://martinfowler.com/articles/mocksArentStubs.html
- Mockito Tutorial
  - https://www.tutorialspoint.com/mockito/index.htm
- Using the Mockito API (section 4 et. seq.)
  - http://www.vogella.com/tutorials/Mockito/article.html
- Chapter 4 from Ammann & Offutt
  - soon to be available at the library via digital reserve.