

How Students Unit Test: Perceptions, Practices, and Pitfalls

GINA R. BAI

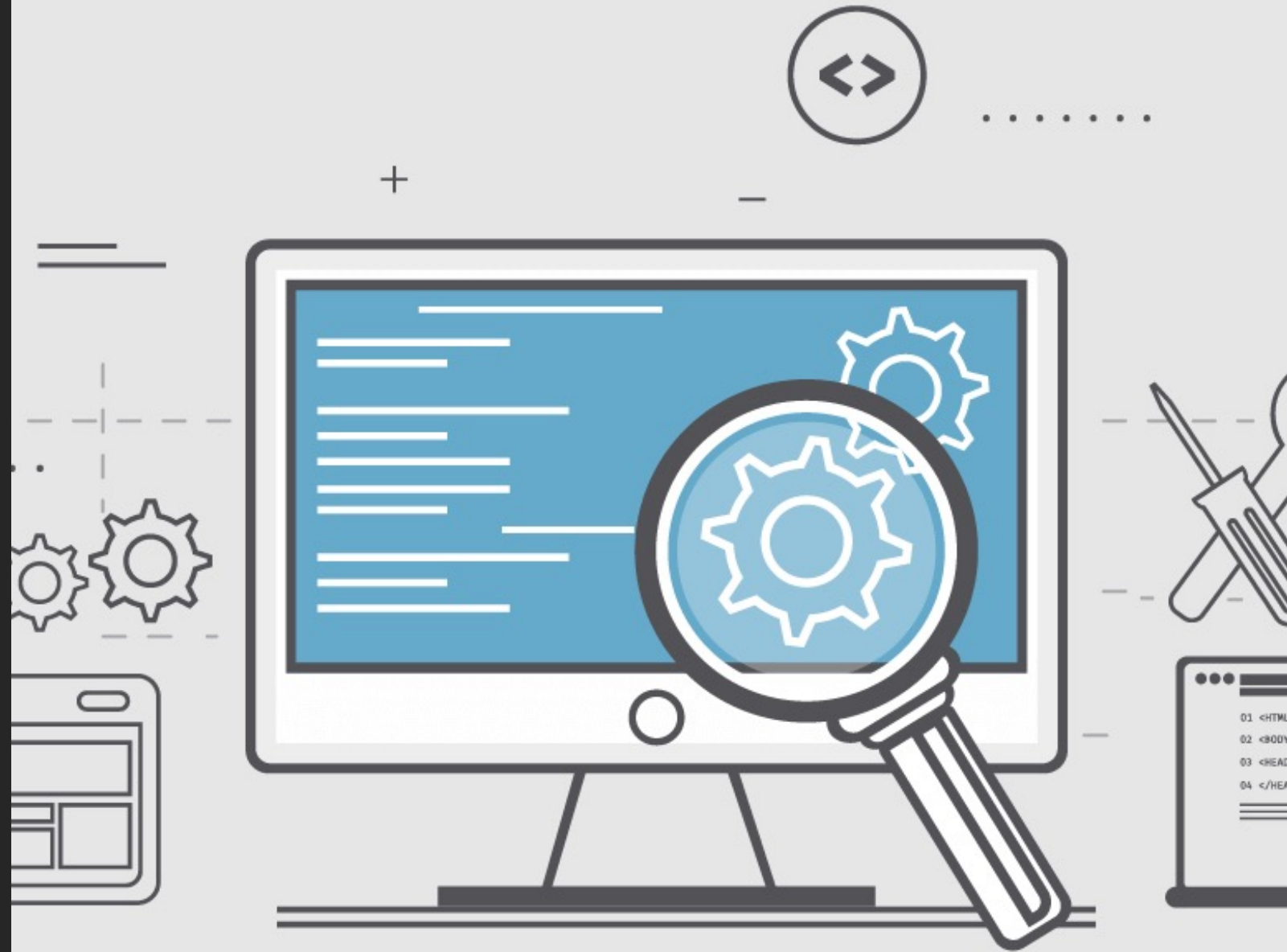
KATHRYN T. STOLEE

NORTH CAROLINA STATE UNIVERSITY

JUSTIN SMITH

LAFAYETTE COLLEGE

JUNE 2021



Testing Education

- ✓ Integrating testing throughout the CS curriculum, from CS1 to postgraduate level courses
[Heckman, ICSTW'20][Janzen, SIGCSE'08][Marrero, ITiCSE'05]...

- ✓ Educational tools and games
[Bradshaw, SIGCSE'15][Elbaum, ICSE'07][Spacco, OOPSLA'06]...

Misconceptions?
Challenges?

Study Design

- ❖ Lab setting, outside the classroom
- ❖ 2 universities, 54 participants
 - ❖ 36 graduates
 - ❖ 18 undergraduates

Study Design

Perception



Q: How do you test your own code?

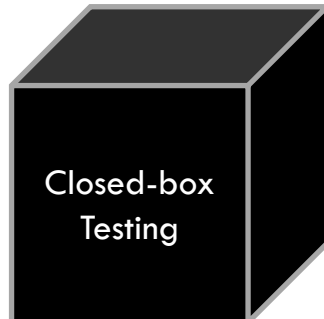
Q: Do you have experience with creating/editing/maintaining unit tests?

- NO
- YES (**37/54**) → Perceptions Questions
 - Motivation
 - Important aspects
 - Techniques
 - Challenges

2-hour lab session

Study Design

Perception



Mars Rover API

- Provided description of expected behaviors
 - Tracks position of a rover and obstacles encountered on a grid
- Implementation-free

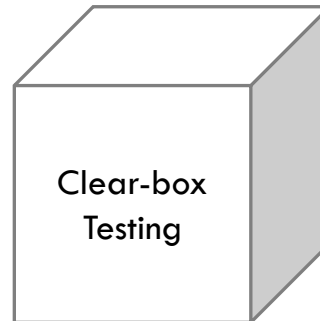
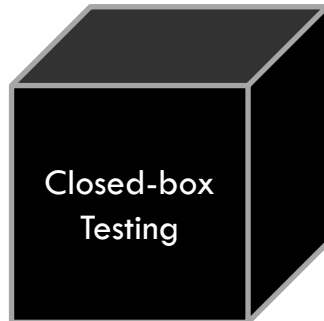
```
@Test
public void theRoverMovesForwardOnceLanding() throws Exception {
    /*
     * Assumption/Scenario: A rover is being created/landed and moved forward based
     * on the string command "f"
     *
     * Testing Input: "f"
     *
     * Expected Output: rover position: (0,1,N)
     */

    fail("Not yet implemented");
}
```

2-hour lab session

Study Design

Perception



Bowling Score Keeper

- Provided description of expected behaviors
 - Calculates the score of a single bowling game
- Provided source code
 - Informed the existence of bugs
- Implementation required (JUnit)

```
@Test
public void testFrameScore() throws Exception {
    Frame f = new Frame(2, 4);
    assertEquals(6, f.score());
}
```

2-hour lab session

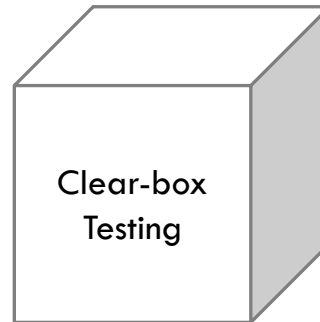
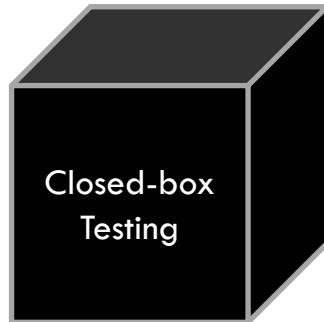
Study Design

Q: What challenge(s) did you encounter when creating/editing the test case?

Q: Demographic Information

- Experience in programming, unit testing
- Prior education in unit testing and software testing

Perception



Post



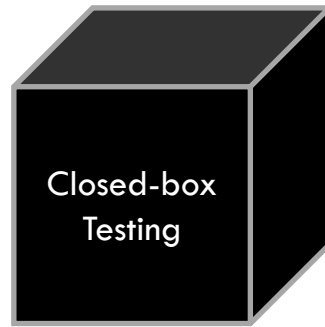
2-hour lab session

Overview of Data

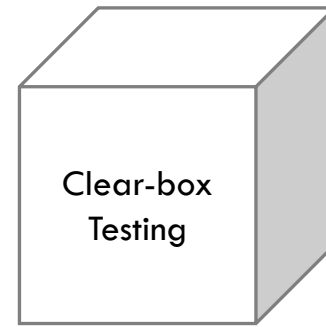
Perception



37
respondents



361 tests
54 students



433 tests
54 students

Post



54
respondents

Research Questions

✓ **Perceptions**

- ✓ RQ1: What standards do students perceive make unit tests good?
- ✓ RQ2: What aspects of unit testing do students perceive to be challenging?

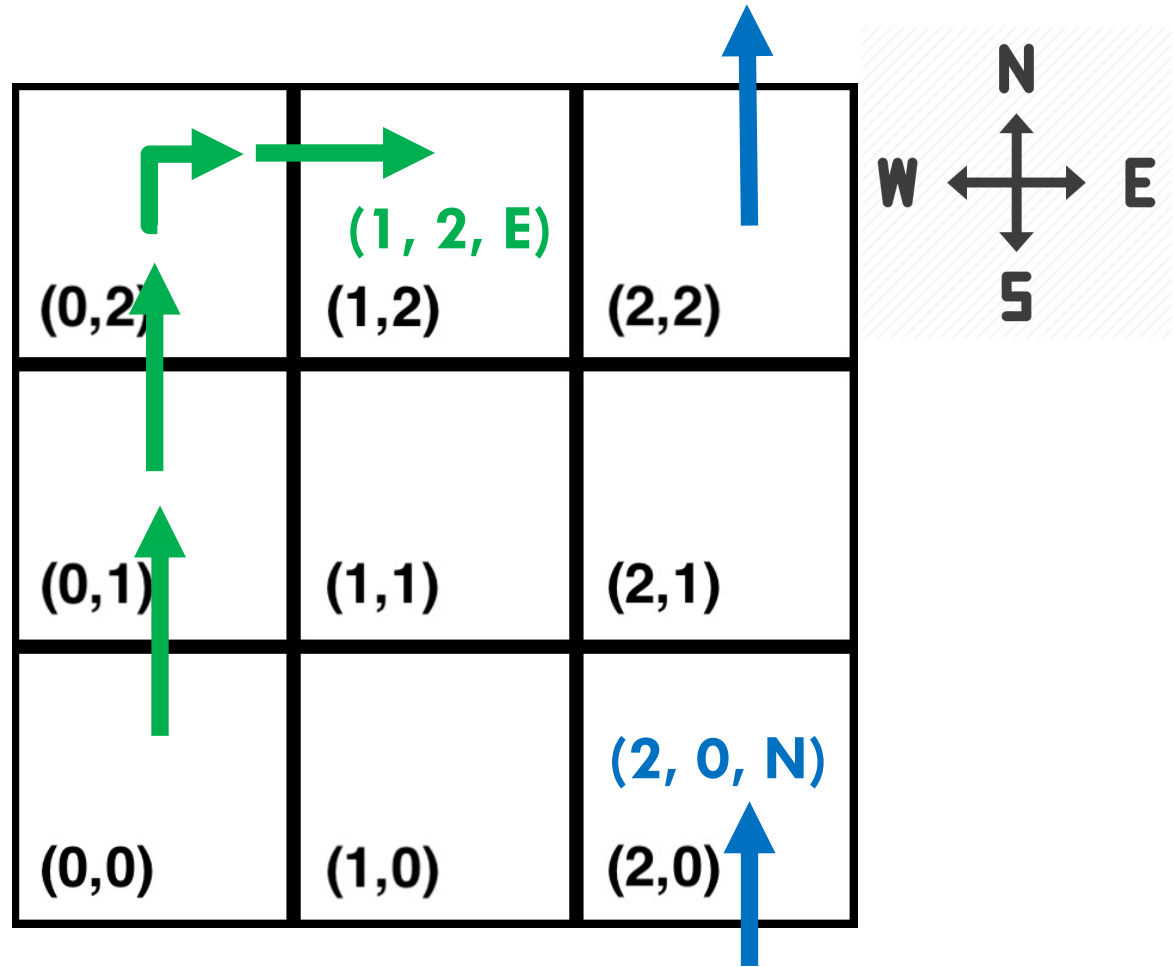
✓ **Practices**

- ✓ RQ3: How well do students perform unit testing?
- ✓ RQ4: What challenges do students encounter when creating or editing tests?

✓ **Pitfalls**

- ✓ RQ5: Does student-written test code smell good?

Example –
Command: f f r f, starting from (0, 0, N)



Closed-box – Mars Rover API

```

11 MarsRover rover;
12
13 @Before
14 public void setUp() throws Exception {
15     rover = new MarsRover(100, 100, "");
16 }
17
18 @Test
19 public void theRoverIsLandedAndExecutedAnEmptyCo
20     /*
21     * Assumption/Scenario: A rover is being cre
22     * command
23     *
24     * Testing Input: NULL
25     *
26     * Expected Output: rover position: (0,0,N)
27     */
28
29     fail("Not yet implemented");
30 }
31
32 @Test
33 public void theRoverMovesForwardOnceLanding() thro
34     /*
35     * Assumption/Scenario: A rover is being creat
36     * on the string command "f"
37     *
38     * Testing Input: "f"
39     *
40     * Expected Output: rover position: (0,1,N)
41     */
42
43     fail("Not yet implemented");
44 }
45

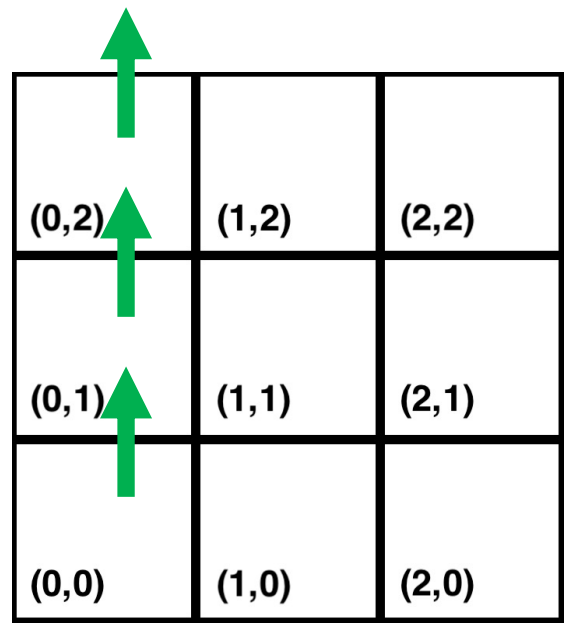
```

**** Good Practices ****

- Descriptive name
- Test if the rover can
 - move forward multiple times
 - move across the upper boundary from South to North

**** Pitfalls/Mistakes ****

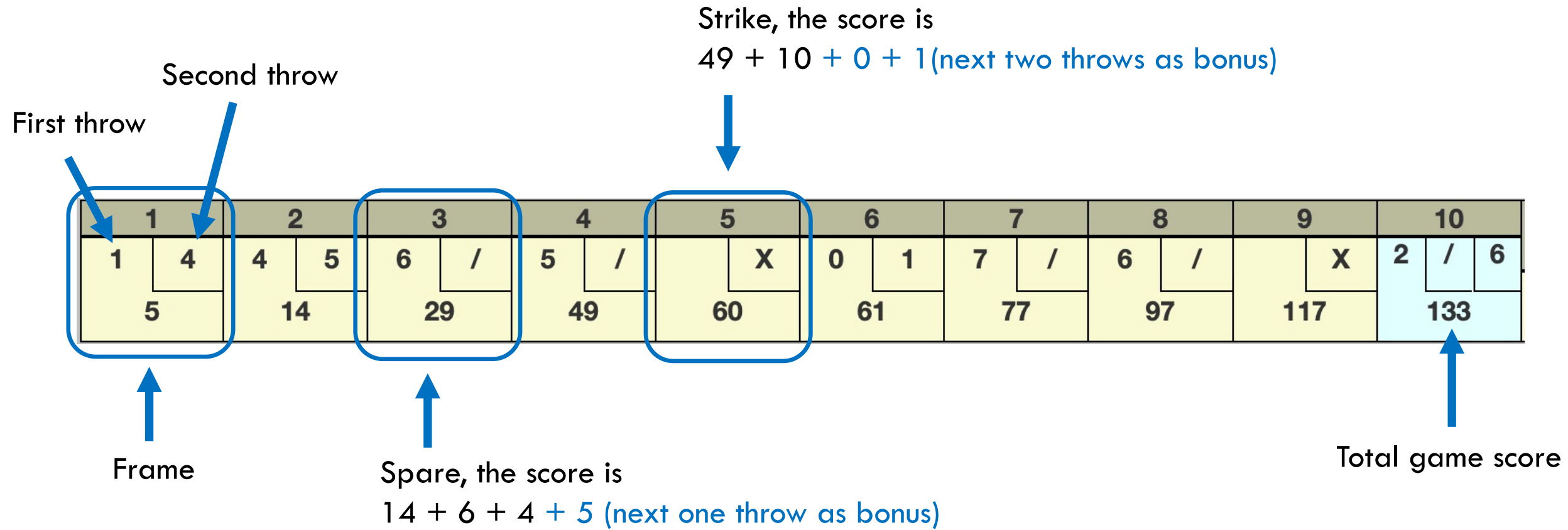
Ignore the setup, 100 x 100 grid size
(Refused Bequest, 37/54)



```

47
48 @Test
49 public void theRoverCircumnavigatesPlanet() throws Exception {
50     /*
51     * Assumption/Scenario: A rover is being created/landed and moved forward based
52     * on the string command "fff"
53     *
54     * Testing Input: "fff"
55     *
56     * Expected Output: rover position: (0,0,N)
57     */
58
59     fail("Not yet implemented");
60 }

```



Clear-box – Bowling Score Keeper

```

@Test(expected = BowlingException.class)
public void testFrameWithScoreIsNotCreated1() throws Exception {
    Frame f = new Frame(-1, 2);
}
@Test(expected = BowlingException.class)
public void testFrameWithScoreIsNotCreated2() throws Exception {
    Frame f = new Frame(10, 1);
}
@Test(expected = BowlingException.class)
public void testFrameWithScoreIsNotCreated3() throws Exception {
    Frame f = new Frame(10, -11);
}
@Test(expected = BowlingException.class)
public void testFrameWithScoreIsNotCreated4() throws Exception {
    Frame f = new Frame(-11, 12);
}

```

```

9 public Frame(int firstThrow, int secondThrow) throws BowlingException {
10     if (firstThrow > 10 || firstThrow < 0
11         || secondThrow > 10 || secondThrow < 0
12         || firstThrow + secondThrow > 10 || firstThrow + secondThrow < 0
13     ){
14         throw new BowlingException();
15     }
16     this.firstThrow = firstThrow;
17     this.secondThrow = secondThrow;
18 }

```

Element	Coverage	Covered Branches	Missed Branches	Total Branches
Task2_BowlingScoreKeeper	92.2 %	59	5	64
src	92.2 %	59	5	64
tdd.bsk	88.6 %	39	5	44
Frame.java	80.0 %	16	4	20
Frame	80.0 %	16	4	20
Frame(int, int)	75.0 %	9	3	12

**** Good Practice ****

1. Test unhappy path
 - 24/54 tested Happy Path Only

**** Can be Improved ****

1. Missing few test cases

```

// return whether the next frame is a bonus frame
public boolean isNextFrameBonus() {
    return frameCounter > 10;
}

@Test
public void testIsNextFrameBonus() throws Exception {
    BowlingGame bg = new BowlingGame();
    for (int k = 0; k < 11; k++) {
        assertFalse(bg.isNextFrameBonus());
        bg.addFrame(new Frame(1,0));
    }
    assertTrue(bg.isNextFrameBonus());
}

```

** Pitfalls/Mistakes **

1. Have **conditional test logic** in test cases
2. The testing scenarios **mismatch** the program specification – 10 frames in a game
(46/54)

** Challenges – Perceived **

1. Understand the source code
2. What to test, and how to test

** Challenges – Unperceived **

1. Understand the program specification
2. Understand the goal of testing

Takeaways

- ❖ Students frequently created test cases that mismatched the program specifications
- ❖ Students will likely ignore setups and only test the happy path when creating unit tests
- ❖ Students are better in designing than implementing

Gina Bai is on the Academic Job Market for a Teaching Position
rbai2@ncsu.edu