

Objectives

- Eclipse debugger
- Software testing issue: when have I tested enough?
- Coverage criteria

Oct 21, 2011

Sprenkle - CSCI209

1

Review

- Describe the general testing process
- What is a set of test cases called?
- What is *unit testing*?
- What are the benefits of unit testing?
- What are the characteristics of good unit tests?
- What are the steps in a JUnit Test Case?
- True or false: When my application passes all of the unit tests, my application is 100% bug free.
- Project 1 questions?

Oct 21, 2011

Sprenkle - CSCI209

2

Eclipse Debugger

1. Set breakpoint
 - Near and BEFORE point of failure
2. Run program in debug mode
3. Inspect variables
4. Step through program, inspecting variables
 - Step into, over, and return

Oct 21, 2011

Sprenkle - CSCI209

3

Software Testing Issues

- How do we know if the calculator program is correct?
 - How do we know that we've exposed all the faults?
 - How confident are we in its correctness?
- How do we know if we've tested enough?
 - Our customers want this product soon but we need product to be correct
 - Harder to fix after it has been released



Oct 21, 2011

Sprenkle - CSCI209

4

Software Testing Issues

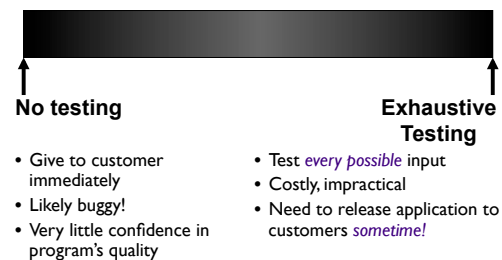
- How do we know if the calculator program is correct?
 - How do we know that we've exposed all the faults?
 - How confident are we in its correctness?
- How do we know if we've tested enough?
 - Time? It's been a couple hours/days/...
 - Number of test cases executed? A lot!
 - I asked my brother and he's really smart and he says that it's enough

Oct 21, 2011

Sprenkle - CSCI209

5

Testing Continuum

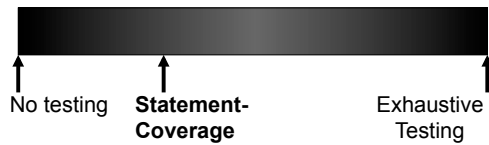


Oct 21, 2011

Sprenkle - CSCI209

6

Testing Continuum



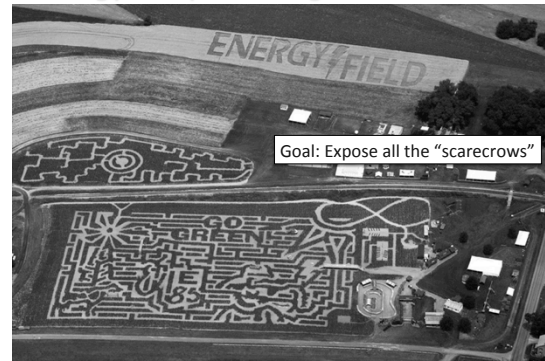
- Need to execute **all code**
- Cover (i.e., execute) all **statements** in the program

Oct 21, 2011

Sprenkle - CSCI209

7

Analogy: Map coverage



Statement Coverage

- Cover all statements in the program

Test Suite:

num=5

```

✓ public String exampleMethod(int num) {
✓   String string = null;
✓   if (num < 10) {
✓     string = "" + condition;
  }
  // remove the leading & trailing whitespace
  return string.trim();
}

```

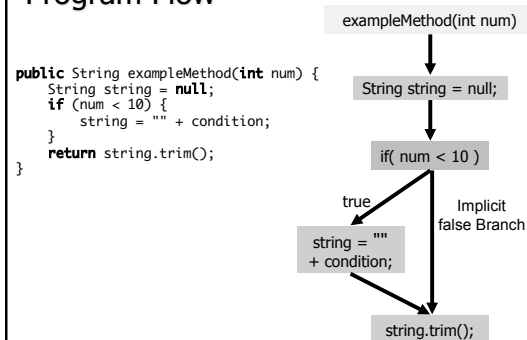
Is this method bug-free?

Oct 21, 2011

Sprenkle - CSCI209

9

Program Flow



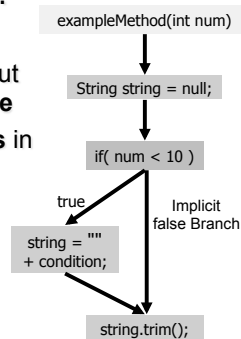
Oct 21, 2011

Sprenkle - CSCI209

10

What Went Wrong?

- Test suite had 100% statement coverage but missed a **branch/edge**
- Try covering all **edges** in program's flow
 - Also covers all **nodes**
 - Called **Branch Coverage**



Oct 21, 2011

Sprenkle - CSCI209

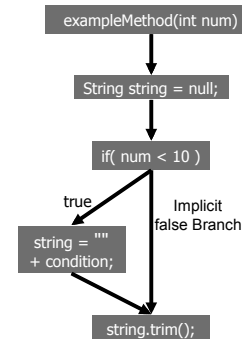
11

Branch Coverage

- Cover all **branches** in the program

Test Suite:

num=5,
num=10



Oct 21, 2011

Sprenkle - CSCI209

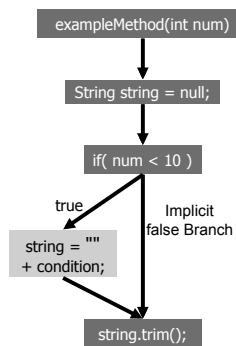
12

Branch Coverage

- Cover all **branches** in the program

Test Suite:

num=5,
num=10



Oct 21, 2011

Sprenkle - CSCI209

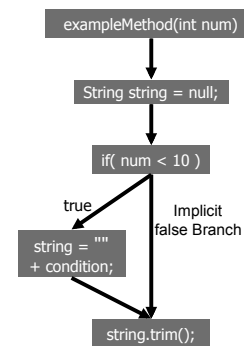
13

Branch Coverage

- Cover all **branches** in the program

Test Suite:

num=5,
num=10



Oct 21, 2011

Sprenkle - CSCI209

14

Example 2

```

public String exampleMethod(int a) {
    String str = "d";
    if ( a < 7 ) {
        a *= 2;
        str += "riv";
    } else {
        str = "co" + str;
    }

    if( a > 10 ) {
        str += "ing";
    } else {
        str += "es";
    }
    return str.substring(6);
}

```

Oct 21, 2011

Sprenkle - CSCI209

15

Example 2

```

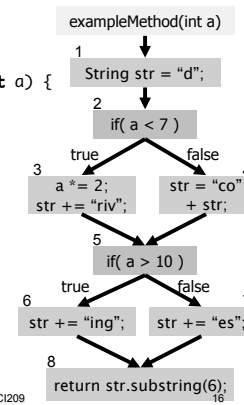
public String exampleMethod(int a) {
    String str = "d";
    if ( a < 7 ) {
        a *= 2;
        str += "riv";
    } else {
        str = "co" + str;
    }

    if( a > 10 ) {
        str += "ing";
    } else {
        str += "es";
    }
    return str.substring(6);
}

```

Oct 21, 2011

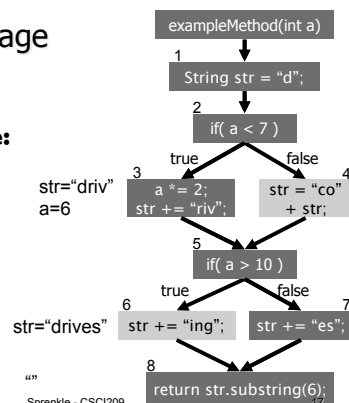
Sprenkle - CSCI209



Branch Coverage

Test Suite:

a=3,
a=30



Oct 21, 2011

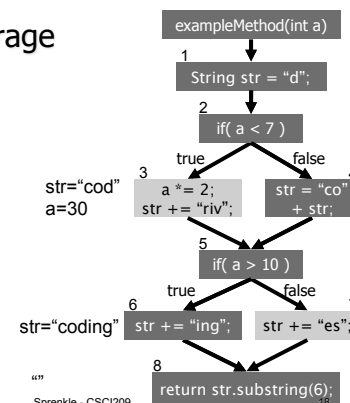
Sprenkle - CSCI209

16

Branch Coverage

Test Suite:

a=3,
a=30



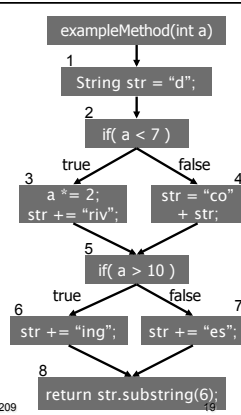
Oct 21, 2011

Sprenkle - CSCI209

17

Branch Coverage

Test Suite:
a=3,
a=30

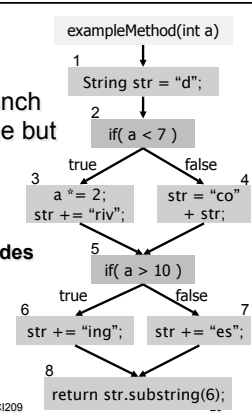


Oct 21, 2011

Sprenkle - CSCI209

What Went Wrong?

- Test suite had 100% branch (and statement) coverage but missed a **path**
- Try to cover all **paths** in program's flow
 - Also gets all **branches, nodes**
 - Called **Path Coverage**

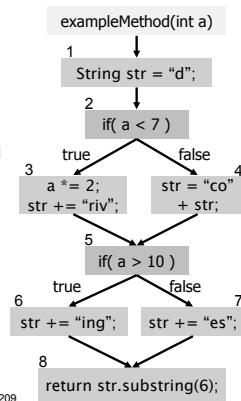


Oct 21, 2011

Sprenkle - CSCI209

Path Coverage

- Cover all **paths** in program's flow
- How many paths through this method?

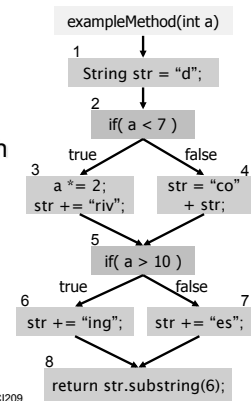


Oct 21, 2011

Sprenkle - CSCI209

Path Coverage

- Cover all **paths** in program's flow
- How many paths through this method?
 - 1-2-3-5-6-8
 - 1-2-3-5-7-8
 - 1-2-4-5-6-8
 - 1-2-4-5-7-8
- What test cases would give us path coverage?

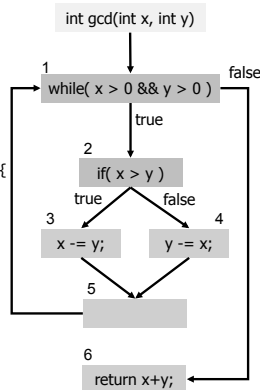


Oct 21, 2011

Sprenkle - CSCI209

Example 3

```
/**
 * Euclid's algorithm to
 * calculate greatest
 * common divisor
 */
public int gcd( int x, int y ) {
    while ( x > 0 && y > 0 ) {
        if ( x > y ) {
            x -= y;
        } else {
            y -= x;
        }
    }
    return x+y;
}
```



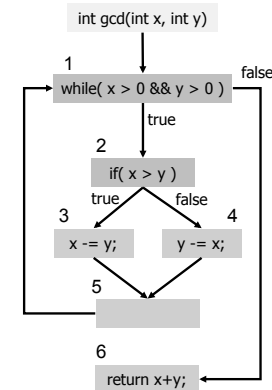
Oct 21, 2011

Sprenkle - CSCI209

Path Coverage

- How many paths through this method?
 - Too many to count, test them all!

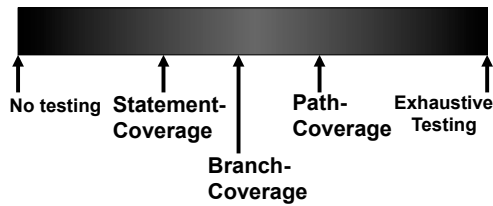
1-6
1-2-3-5-1-6
1-2-4-5-1-6
1-2-3-5-1-2-3-5-1-6
1-2-4-5-1-2-4-5-1-6
1-[2-(3|4)-5-1]*-6



Oct 21, 2011

Sprenkle - CSCI209

Testing Continuum



Oct 21, 2011

Sprenkle - CSCI209

25

Comparison of Coverage

Coverage Criterion	Advantages	Disadvantages
Statement		
Branch		
Path		

Oct 21, 2011

Sprenkle - CSCI209

26

Comparison of Coverage

Coverage Criterion	Advantages	Disadvantages
Statement	Practical	Weak, may miss many faults
Branch	Practical, Stronger than Statement	Weaker than Path
Path	Strongest	Infeasible, too many paths to be practical

Oct 21, 2011

Sprenkle - CSCI209

27

Uses of Coverage Criteria

- "Stopping" rule → sufficient testing
 - Avoid unnecessary, redundant tests
- Measure test quality
 - Dependability estimate
 - Confidence in estimate
- Specify test cases
 - Describe additional test cases needed

Oct 21, 2011

Sprenkle - CSCI209

28

Coverage Criteria Discussion

- Is it always possible for a test suite to cover all the statements in a given program?
 - No. Could be infeasible statements
 - Unreachable code
 - Legacy code
 - Configuration that is not on site
- Do we need the test suite to cover 100% of statements/branches to believe it is adequate?
 - 100% coverage does not mean correct program
 - But < 100% coverage does mean testing inadequacy

Oct 21, 2011

Sprenkle - CSCI209

29

Looking Ahead

- Monday
 - Coverage tools, Design principles
- Wednesday
 - Project 1 due
- Extra Credit Opportunities
 - Friday, November 4, Turing Award in CS (10 pts)
 - Commons – Women's Resource Room
 - Professor Stough to present on prize awarded to Leslie G. Valiant
 - Learning a Java API: Regular Expressions (up to 50)
 - See web site, due by last day of class

Oct 21, 2011

Sprenkle - CSCI209

30