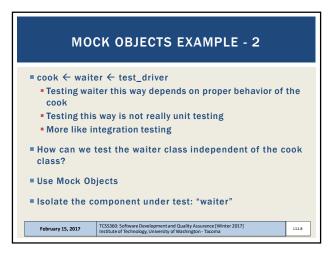
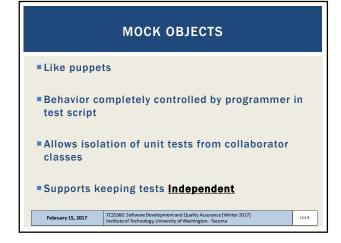
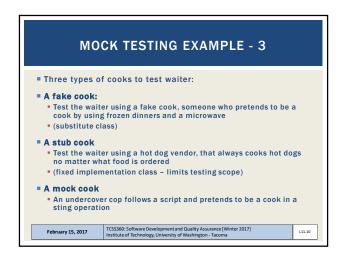
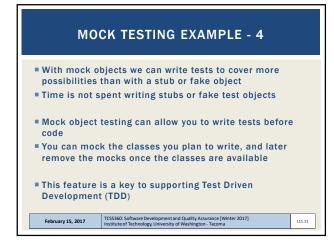


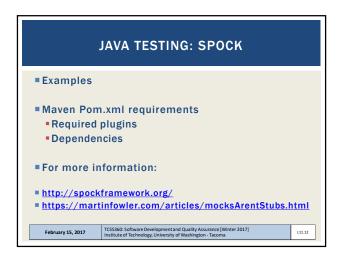
# MOCK OBJECTS - EXAMPLE Consider unit testing for a class hierarchy: cook ← waiter ← customer Testing cook is easy since it has no dependencies cook ← test\_driver We can write simple tests to exercise the methods of cook Simpler to data objects (e.g. user class)



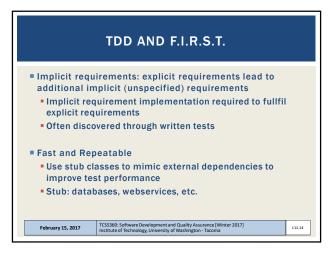












## TEST COVERAGE How much testing is enough? As much as you can do before the shipping deadline Code-to-test ratio (non-comment lines of code) Production < 1: more test code than app code Types of tests Integration/system tests Functional tests TCS360: Software Development and Quality Assurance [Winter 2017] Institute of Technology, University of Washington - Tacoma

```
TEST COVERAGE: SAMPLE CODE
    Public class MyClass
2
    -{
       int w;
3
       public void food(int x, int y, int z) {
          if (x)
            if (y && z)
               bar(0);
          else
9
            bar(1);
10
11
       public void bar(int x) {
12
         W = X;
13
      3
14 }
                   TCSS360: Software Development and Quality Assurance [Winter 2017]
Institute of Technology, University of Washington - Tacoma
  February 15, 2017
                                                                      L11.16
```

```
Code coverage
Percentage of code exercised by testing
Not always straightforward to measure
Level are increasingly difficult to achieve

Method coverage (S0) - Is every method executed at least once?

Must call foo and bar at least once

Call coverage (S1) - Has every method been called from every place it could be called?
Requires calling bar from line 7 & 9

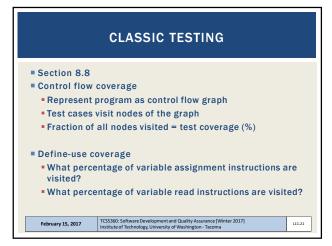
TCSS60: Software Development and Quality Assurance [Winter 2017] Institute of Technology, University of Washington - Taccoma

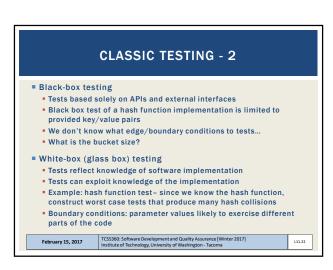
L11.17
```

### **LEVELS OF TEST COVERAGE - 2** ■ Statement coverage (CO) - Is every statement of the source code executed at least once by the test suite? > Count each branch of a conditional as a single statement: must call with x=true, y=false ■ Branch coverage (C1) - Has every branch been taken at least once? > Must call foo with x=true&false and also y&z so both branches execute Subset - Decision coverage: Each subexpression that affects a conditional expression must be tested with both a true and false value. y&z where y=false, y&z where z=false Path coverage (C2) - Has every possible route through the code been executed? > If x,y,z are booleans, then there are 8 possible paths February 15, 2017 TCSS360: Software Development and Quality Assurance [Win Institute of Technology, University of Washington - Tacoma L11.18

# TEST COVERAGE - 3 Achieving statement coverage (C0) is straightforward Achieving branch coverage (C1) is more difficult: Test cases must ensure each branch is taken at least once in each direction Path coverage (C2) is very difficult: Not all experts agree on the value of path coverage Does high test coverage imply a well-tested application? Coverage says nothing about the quality of the tests. Low coverage implies a poorly tested application

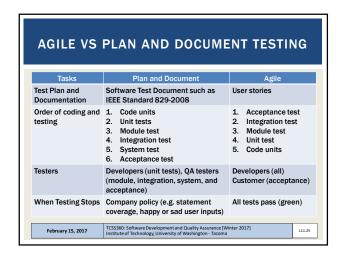
# TEST COVERAGE - 4 Code coverage statistics Highlight under-tested, untested parts of application code Show the overall comprehensiveness of test suite A number of tools exist to evaluate code coverage of tests For Java see Jcov, Clover, EMMA, Serenity TCSSSGO Software Development and Quality Assurance [Winter 2017] Institute of Technology, University of Washington Taxoms

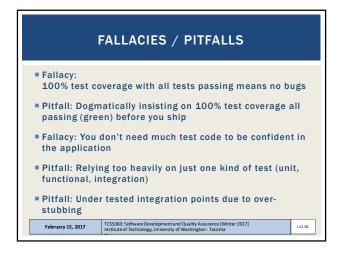


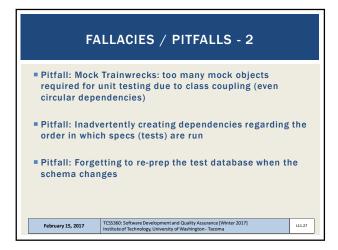


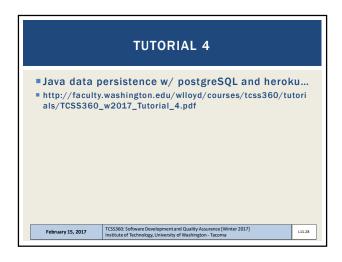
### 

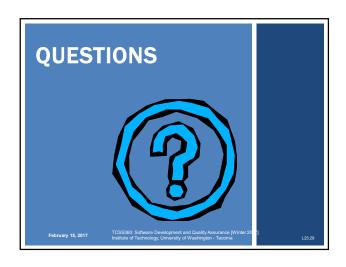
### PLAN AND DOCUMENT PERSPECTIVE Composing unit tests for integration testing: Top-down integration Bottom-up integration Sandwich integration Formal testing methods - use formal specification to prove the behavior of the code using mathematical proofs Automatic theorem proving Model checking Verify selected properties via exhaustive search of all possible states a system could enter during execution February 15, 2017 Tissib: Software Developmentad Quality Assurance [Winter 2017] Institute of Technology, University of Washington-Tacoma

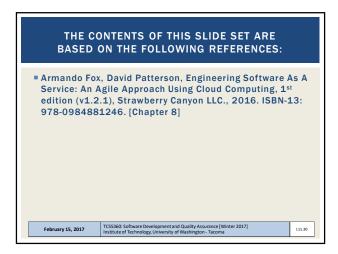












```
SPOCK SPEC SAMPLE

def "HashMap accepts null key"() {
    setup:
    def map = new HashMap()

    when:
    map.put(null, "elem")

    then:
    notThrown(NullPointerException)
}
```

```
def "events are published to all subscribers"() {
    def subscriber1 = Mock(Subscriber)
    def subscriber2 = Mock(Subscriber)
    def publisher = new Publisher()
    publisher.add(subscriber1)
    publisher.add(subscriber2)

when:
    publisher.fire("event")

then:
    1 * subscriber1.receive("event")
    1 * subscriber2.receive("event")
}

February 15, 2017

TCSSSGO. Software Development and Quality Assurance [Winter 2017]
    Institute of Technology, University of Washington-Tacoma
```

```
SPOCK SAMPLE SPEC - 3

def "offered PC matches preferred configuration"() {
    when:
    def pc = shop.buyPc()

    then:
    with(pc) {
        vendor == "Sunny"
        clockRate >= 2333
        ram >= 406
        os == "Linux"
    }
}
```