

# CS239

Nathan Sprague

February 22, 2012

## Reading Quiz (1/3)

Which of the following statements is most accurate?

- 1 JUnit automates the process of designing tests: it analyzes source code and generates an appropriate set of tests.
- 2 JUnit automates the testing process. After a programmer develops the tests, JUnit is used to run the tests and report the results.
- 3 When used correctly, JUnit can provide strong guarantees about the correctness of tested code.

## Reading Quiz (2/3)

What is the meaning of `@Test` in the context of JUnit testing?

- 1 `@Test` is an annotation that indicates that a method should be executed as a JUnit test.
- 2 `@Test` is a special Java method that returns true if the following method performs correctly.
- 3 `@Test` is a special JavaDoc symbol used to document the fact that a method still needs to be tested.

## Reading Quiz (3/3)

How many times is the method `setUp` executed when the following class is executed as a JUnit test suite?

```
1
2 public class AccountTest {
3
4     private Account account;
5
6     @Before public void setUp() {
7         account = new Account("Bob", "100 Drury Lane");
8     }
9
10    // Remaining test code not shown...
```

- 1 Once.
- 2 Before each test method is executed.
- 3 Never.

# Different Perspectives

```
1      /*****
2      * Return the number of lines of text in this document.
3      *
4      * @return Number of lines.
5      *****/
6      public int numLines()
7      {
8          int total = 0;
9
10         for (int i = 0; i < words.length; i++)
11         {
12             total++;
13         }
14         return total;
15     }
```

- 1 Perfectly good solution.
- 2 Returns correct value. Design could be better, but it isn't a big deal.
- 3 Returns correct value. Design could be better and it *is* a big deal.
- 4 Returns incorrect value.

# Which is the best design? (D if there is a functional difference.)

```
1 public static boolean containsA(int[] numbers, int number) {
2     boolean found = false;
3     for (int i = 0; i < numbers.length; i++) {
4         if (numbers[i] == number) {
5             found = true;
6             i = numbers.length + 1;
7         }
8     }
9     return found;
10 }
11
12 public static boolean containsB(int[] numbers, int number) {
13     boolean found = false;
14     for (int i = 0; i < numbers.length; i++) {
15         if (numbers[i] == number) {
16             found = true;
17             break;
18         }
19     }
20     return found;
21 }
22
23 public static boolean containsC(int[] numbers, int number) {
24     boolean found = false;
25     for (int i = 0; i < numbers.length && !found; i++) {
26         if (numbers[i] == number) {
27             found = true;
28         }
29     }
30     return found;
31 }
```