Name: Solutions

Chapter 8 Quiz

1. [2pts] TDD Rule #1: Your test should always _____ before you implement any code.

   Fail

2. [4pts] TDD Rule #2: Implement the _____(a)_____ CODE _____(b)_____ to make your tests pass.

   (a) simplest (b) possible

3. [3pts] What does the YAGNI principle stand for?

   You Aren't Gonna Need It

4. [2pts] Which order is correct?

   a. green, red, refactor

   b. green, refactor, red

   (c.) red, green, refactor

   d. red, refactor, green

   e. None of the above

5. [2pts] _____ coupled systems are brittle and difficult to maintain, not to mention really, really hard to test.

   a. Loosely

   (b.) Tightly

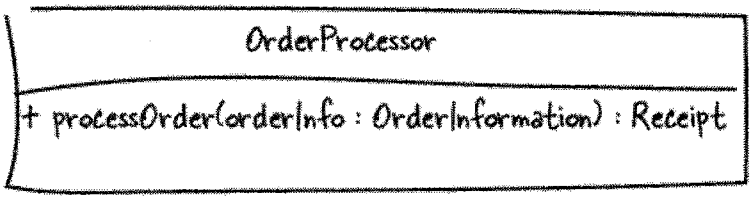
   c. None of the above

6. [2pts] T or F. Each test should verify only one thing. Even if the test case tests multiple methods, it still should test only one piece of functionality.

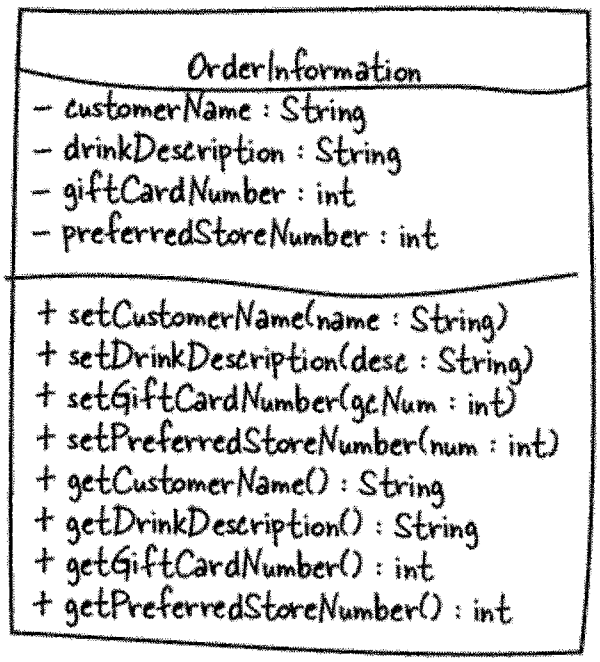
   T

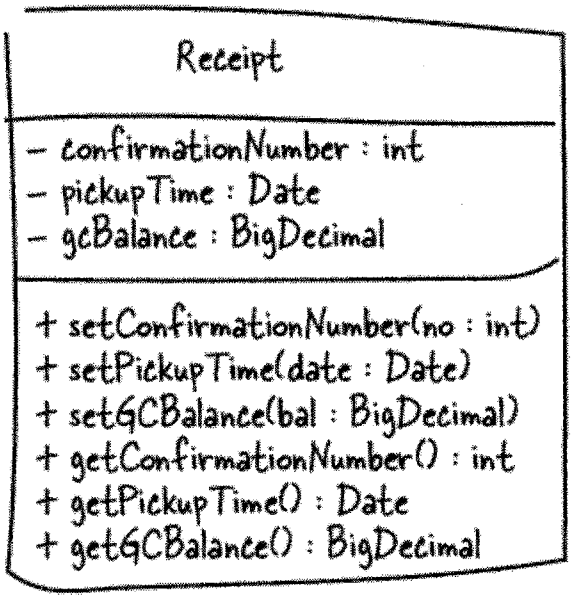
7. [10pts] Implement a JUnit test that will verify your software can process a simple order.

Assume that this is what the class that processes the orders will be like:



And here are two existing classes:





From JUnit, use only the **fail** method:

fail

```
public static void fail(java.lang.String message)
```

Fails a test with the given message.

Parameters:

`message` - the identifying message for the `AssertionError` (`null` okay)

Write your answer on the next page. I've started the test code for you.

```
@Test
public void testSimpleOrder() {
    OrderProcessor op = new OrderProcessor();

    OrderInformation oi = new OrderInformation();
    oi.setCustomerName("Dan");
    oi.setDrinkDescription("Bold with room");
    oi.setGiftCardNumber(12345);
    oi.setPreferredStoreNumber(123);

    Receipt r = op.processOrder(oi);

    if (r == null) {
        Fail("Doh! processOrder returned null.");
    }
    if (r.getConfirmationNumber() <= 0) {
        fail("Bad confirmation number.");
    }
    if (!r.getGCBalance().equals(0)) {
        fail("Bad GC balance:");
    }

}
```

(I didn't need this page.)