

Testing in Practice

Software Testing
(3104313)

Amirkabir University of Technology
Spring 1399-1400

Test Implementation

Theory is usually further from practice than we wish!

Integration Testing

- Testing of **incompatibilities** and **interfaces** between otherwise *correctly working* components.

?? Integration Order

?? Incomplete Systems

Class Integration Test Order (CITO)

The general **goal** is to integrate and test classes in the order that requires the **least scaffolding**, or additional software.

Class Integration Test Order (CITO)

The general **goal** is to integrate and test classes in the order that requires the **least scaffolding**, or additional software.

Test Doubles

- Any kind of **pretend** object used in place of a real object for testing purposes.
 - Implements **partial** functionality.
 - Mocks, Stubs,
-
- The components have **not yet been written**,
 - The components do something that we **can't afford** to happen during testing.

Tests with Stubs

```
public interface MailService {
    public void send (Message msg);
}

public class MailServiceStub implements MailService {
    private List<Message> messages = new ArrayList<Message>();
    public void send (Message msg) {
        messages.add(msg);
    }
    public int numberSent() {
        return messages.size();
    }
}

public void testOrderSendsMailIfUnfilled() {
    Order order = new Order(TALISKER, 51);
    MailServiceStub mailer = new MailServiceStub();
    order.setMailer(mailer);
    order.fill(warehouse);
    assertEquals(1, mailer.numberSent());
}
```

```
public void testFillingDoesNotRemoveIfNotEnoughInStock() {  
    Order order = new Order(TALISKER, 51);  
    Mock warehouse = mock(Warehouse.class);  
  
    warehouse.expects(once()).method("hasInventory")  
        .withAnyArguments()  
        .will(returnValue(false));  
  
    order.fill((Warehouse) warehouse.proxy());  
  
    assertFalse(order.isFilled());  
}
```

Tests with
Mock Objects

How are test doubles integrated and managed?