

# In-Class Exercises Week 5

**Software Testing**

SWE 437/637

[go.gmu.edu/SoftwareTestingFall24](https://go.gmu.edu/SoftwareTestingFall24)

**Dr. Brittany Johnson-Matthews**

(Dr. B for short)

# Class Activity #1

---

## Test doubles

Consider testing a class for airline reservation services. It has a ranking service as a dependency, with the idea that higher ranking customers get better arrangements (better seats, early boarding, etc.). *We are not testing the ranking service*, and we should assume it's stochastic (that is, it might give different answers every time).

# A Simple Example

---

An implementation of ReservationService might look like this:

```
public class ReservationService {  
    // instance variables, constructors, other methods omitted for now  
  
    public void reserve (Customer customer) {  
        RankingService rankingService = new RankingService();  
        // more code that uses the ranking service by calling its method  
        //   public Rank getRank(Customer customer)  
        //   on the rankingService object.  
    }  
}
```

An associated test might look like this:

```
@Test public void testReservationService () {  
    ReservationService reservationService = new ReservationService();  
    RankingService fakeRankingService = new FakeRankingService(); // inherits from RankingService  
    // Umm... How do I get RankingService.reserve() to use this test double?  
    // some assertion about reservationService.reserve("John Smith");  
    // some assertion about reservationService.reserve("Jane Doe");  
}
```

# Using the test double

---

What if we make rankingService a member variable of ReservationService?

```
public class ReservationService {  
    private RankingService rankingService;  
  
    // instance variables, constructors, other methods omitted for now  
  
    public void reserve (Customer customer) {  
        rankingService = new RankingService();  
        // more code that uses the ranking service by calling  
        //   public Rank getRank(Customer customer)  
        //   on the rankingService object.  
    }  
}
```

What else do we need to do? Hint: think about **seams**

**Develop one or more simple tests with FakeRankingService.  
Focus on what you want to test, not the JUnit syntax**

# Using Seams

---

Given what we know about seams, develop an approach to using a fake ranking service

1. Using a compiler seam
2. Using an inheritance seam without dependency injection
3. Using an inheritance seam with dependency injection
4. Using Jmockit (try this one if your group finishes your assigned approach)