

# Characterization Testing Kata

Week 4: Practical Session

Advanced Software Engineering

# Learning Objectives

- Practice writing characterization tests for legacy code
- Identify and create seams for testing
- Safely document and preserve existing behavior

# Kata Instructions

Work in pairs to add tests to the following legacy code examples.

## C++ Starting Point

---

```
1  class DiscountCalculator {
2  public:
3      double calculate(double price, int quantity) {
4          double total = price * quantity;
5          if (quantity > 10) {
6              total *= 0.95; // Bulk discount
7          }
8          if (total > 1000) {
9              total *= 0.90; // Large order discount
10         }
11         return total;
12     }
13 };
```

---

## Python Starting Point

---

```
1  class ShippingCostCalculator:
2      def calculate(self, weight, destination):
3          cost = 5.0
4          if destination == "North America":
```

# Tasks

## 1. Write Characterization Tests (30 min)

- Write tests that capture the current behavior for a variety of inputs
- Document any surprising or unclear behavior

## 2. Identify Seams (20 min)

- Find places where you can introduce tests without changing logic
- Note any dependencies that make testing difficult

## 3. Refactor for Testability (30 min)

- Make minimal changes to enable more testing (e.g., extract methods, parameterize dependencies)
- Add additional tests as needed

## 4. Review and Discuss (20 min)

- Share your tests and findings with another pair
- Discuss strategies for handling untestable code

## Success Criteria

- Comprehensive set of characterization tests
- Identification of seams and testability issues
- Minimal, safe refactoring to enable testing
- Clear documentation of current behavior

## Notes for Facilitators

- Encourage participants to avoid changing logic
- Focus on capturing, not correcting, behavior
- Discuss the value of approval testing for complex outputs

## Follow-up

Prepare for next week's session on:

- Safe, non-invasive changes
- Sprouting and wrapping techniques
- Planning incremental improvements