// File: src/test/java/com/example/OrderProcessorTest.java

package com.example;

import static org.junit.Assert.\*;

import static org.mockito.Mockito.\*;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.mockito.InjectMocks;

import org.mockito.Mock;

import org.mockito.junit.MockitoJUnitRunner;

@RunWith(MockitoJUnitRunner.class)

public class OrderProcessorTest {

/\* ---------- External dependency to mock ---------- \*/

interface PaymentGateway {

/\*\* Charges the amount; returns a receipt ID or throws. \*/

String charge(double amount);

}

/\* ---------- System under test -------------------- \*/

static class OrderProcessor {

private final PaymentGateway gateway;

OrderProcessor(PaymentGateway gateway) {

this.gateway = gateway;

}

/\*\* Attempts to pay; returns the receipt ID from the gateway. \*/

String pay(double amount) {

return gateway.charge(amount);

}

}

/\* ---------- Mockito wiring ----------------------- \*/

@Mock

private PaymentGateway gateway; // will be a mock object

@InjectMocks

private OrderProcessor processor; // mock injected automatically

/\* ---------- Test case using Arrange‑Act‑Assert ---- \*/

@Test

public void pay\_shouldReturnReceiptId() {

// Arrange – stub the gateway’s response

when(gateway.charge(42.00)).thenReturn("RCPT-123");

// Act

String receipt = processor.pay(42.00);

// Assert

assertEquals("RCPT-123", receipt);

// And verify interaction (optional but good practice)

verify(gateway).charge(42.00);

verifyNoMoreInteractions(gateway);

}

}