**Mockito Exercises**

**Exercise 1: Verifying Interactions**

**EmailService.java**

**Description:** A dependency we will mock. Represents an email-sending component.

package com.example.verify;

public interface EmailService {

void sendWelcomeEmail(String email);

}

**UserRegistrationService.java**

**Description:** Business logic that registers a user and sends a welcome email.

package com.example.verify;

public class UserRegistrationService {

private EmailService emailService;

public UserRegistrationService(EmailService emailService) {

this.emailService = emailService;

}

public void registerUser(String email) {

// Simulate saving user (omitted)

emailService.sendWelcomeEmail(email);

}

}

**UserRegistrationServiceTest.java**

**Description:** Mockito test that verifies the email was sent exactly once with the correct email address.

package com.example.verify;

import static org.mockito.Mockito.\*;

import org.junit.Test;

public class UserRegistrationServiceTest {

@Test

public void testSendWelcomeEmailCalled() {

// Create mock of EmailService

EmailService mockEmailService = mock(EmailService.class);

// Inject mock into UserRegistrationService

UserRegistrationService registrationService = new UserRegistrationService(mockEmailService);

// Perform action

registrationService.registerUser("keerthi@example.com");

// Verify that sendWelcomeEmail() was called once with expected argument

verify(mockEmailService, times(1)).sendWelcomeEmail("keerthi@example.com");

}

}

**OUTPUT:**

