**Exercise 3: Argument Matching**   
Scenario:  
You need to verify that a method is called with specific arguments.   
Code:  
Create a interface   
NotificationService.java-

package com.Mockito.MocitoArgumentMatching;

public interface NotificationService {

void sendNotification(String user,String msg);

}

Add required dependencies to pom.xml file  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.Mockito</groupId>

<artifactId>MocitoArgumentMatching</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>MocitoArgumentMatching</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

<version>5.12.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-api</artifactId>

<version>5.9.3</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

NotificationServiceTest.java

package com.Mockito.MocitoArgumentMatching;

import org.junit.jupiter.api.Test;

import static org.mockito.Mockito.\*; // Import Mockito static methods

import static org.mockito.ArgumentMatchers.\*; // For eq(), anyString()

public class NotificationServiceTest {

*@Test*

void testSendNotification() {

// Step 1: Create mock

NotificationService service = *mock*(NotificationService.class);

// Step 2: Call method with arguments

service.sendNotification("monalisa", "welcome to the course");

// Step 3: Verify interaction with argument matchers

*verify*(service).sendNotification(*eq*("monalisa"), *anyString*());

}

}

Output: