WEEK 2

JUnit, Mockito and SL4J

3. Setting Up Junit

Created a new Java project in your IDE – Week2

Added JUnit dependency to project

pom.xml:

```
<dependency>
<groupld>junit</groupld>
<artifactld>junit</artifactld>
<version>4.13.2</version>
<scope>test</scope>
</dependency>
```

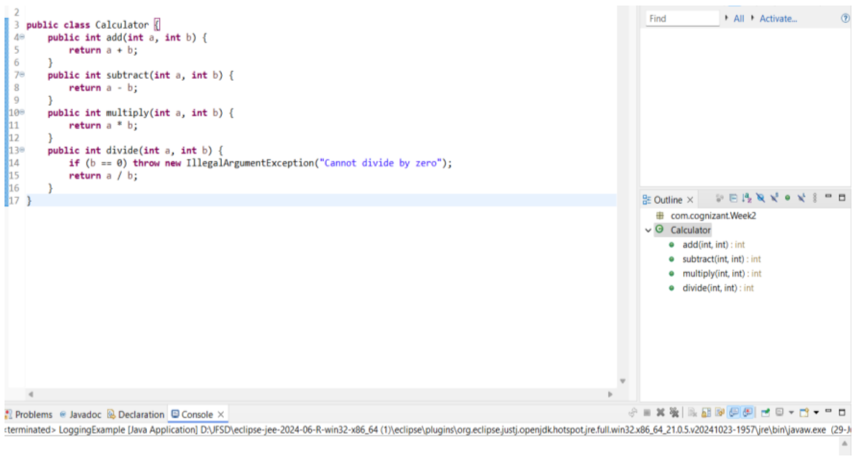
Created a class Calculator.java:

```
package week2;


public class Calculator {
   public int add(int a, int b) {
      return a + b;
   }
   public int subtract(int a, int b) {
      return a- b;
   }
   public int multiply(int a, int b) {
      return a * b;
   }
   public int divide(int a, int b) {
      if (b == 0) throw new IllegalArgumentException("Cannot divide by zero");
      return a / b;
   }
}
```

Created CalculatorTest.java to Test :

```
package week2;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.fail;

import org.junit.Test;

public class CalculatorTest {

	@Test
	public void test() {
		fail("Not yet implemented");
	}
	Calculator calc = new Calculator();

    @Test
    public void testAdd() {
        assertEquals(5, calc.add(2, 3));
    }

    @Test
    public void testSubtract() {
        assertEquals(1, calc.subtract(4, 3));
    }

    @Test
    public void testMultiply() {
        assertEquals(12, calc.multiply(3, 4));
    }

    @Test
    public void testDivide() {
        assertEquals(5, calc.divide(10, 2));
    }

    @Test(expected = IllegalArgumentException.class)
    public void testDivideByZero() {
        calc.divide(10, 0);
    }
}
```

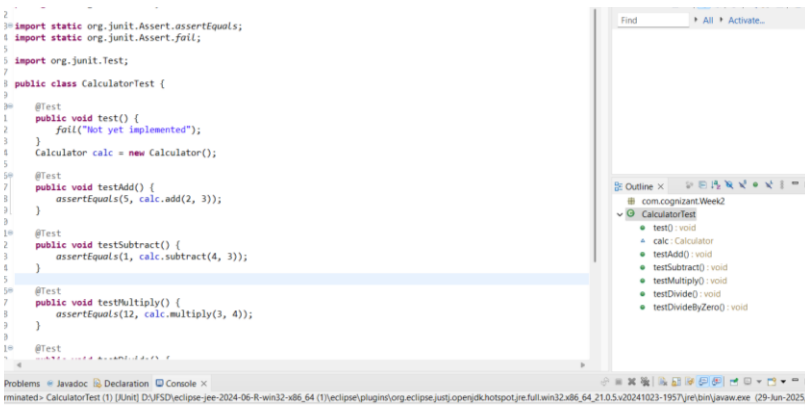
Output:



```
public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }
    public int subtract(int a, int b) {
        return a - b;
    }
    public int multiply(int a, int b) {
        return a * b;
    }
    public int divide(int a, int b) {
        if (b == 0) throw new IllegalArgumentException("Cannot divide by zero");
        return a / b;
    }
}
```

Outline
com.cognizant.Week2
Calculator
- add(int, int) : int
- subtract(int, int) : int
- multiply(int, int) : int
- divide(int, int) : int

```
<terminated> LoggingExample [Java Application] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe (29-J
```



```
import static org.junit.Assert.assertEquals;
import static org.junit.Assert.fail;

import org.junit.Test;

public class CalculatorTest {

    @Test
    public void test() {
        fail("Not yet implemented");
    }
    Calculator calc = new Calculator();

    @Test
    public void testAdd() {
        assertEquals(5, calc.add(2, 3));
    }

    @Test
    public void testSubtract() {
        assertEquals(1, calc.subtract(4, 3));
    }

    @Test
    public void testMultiply() {
        assertEquals(12, calc.multiply(3, 4));
    }

    @Test
```

Outline
com.cognizant.Week2
CalculatorTest
- test() : void
- calc : Calculator
- testAdd() : void
- testSubtract() : void
- testMultiply() : void
- testDivide() : void
- testDivideByZero() : void

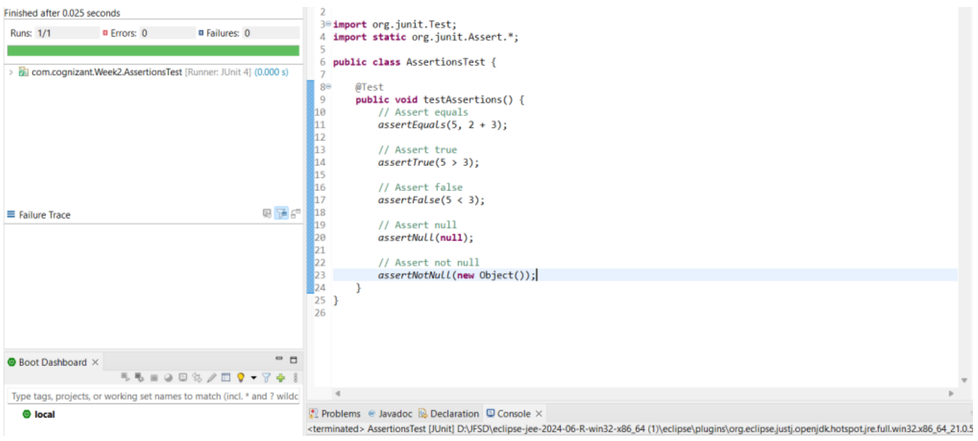
```
rminated> CalculatorTest (1) [JUnit] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe (29-Jun-2025
```

4. Assertions in Junit:-

Created a test file named with AssertionsTest.java.

```
package week2;

import org.junit.Test;
import static org.junit.Assert.*;

public class AssertionsTest {

  @Test
  public void testAssertions() {
    // Assert equals
    assertEquals(5, 2 + 3);

    // Assert true
    assertTrue(5 > 3);

    // Assert false
    assertFalse(5 < 3);

    // Assert null
    assertNull(null);

    // Assert not null
    assertNotNull(new Object());
  }
}
```

Output:



```
Finished after 0.025 seconds
Runs: 1/1    Errors: 0    Failures: 0
> com.cognizant.Week2.AssertionsTest [Runner: JUnit 4] (0.000 s)
Failure Trace
```

```java

import org.junit.Test;
import static org.junit.Assert.*;

public class AssertionsTest {

    @Test
    public void testAssertions() {
        // Assert equals
        assertEquals(5, 2 + 3);

        // Assert true
        assertTrue(5 > 3);

        // Assert false
        assertFalse(5 < 3);

        // Assert null
        assertNull(null);

        // Assert not null
        assertNotNull(new Object());
    }
}

```

```
<terminated> AssertionsTest [JUnit] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5
```

Boot Dashboard
Type tags, projects, or working set names to match (incl. * and ? wildc
local
Problems
Javadoc
Declaration
Console

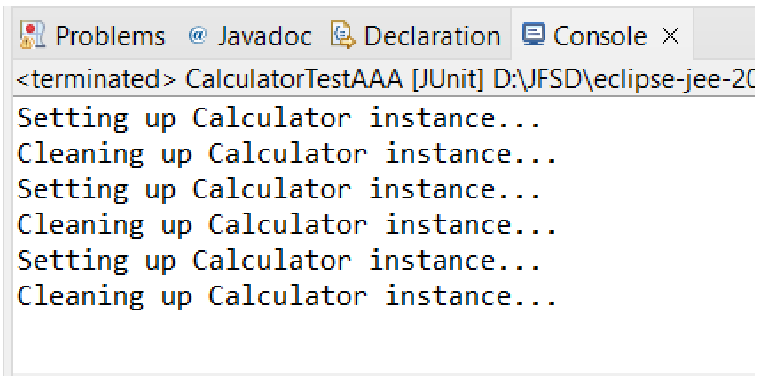
5. Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in Junit

Create a test file named with CalculatorTestAAA.java.

```java
package week2;


import static org.junit.Assert.*;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;


public class CalculatorTestAAA {


    private Calculator calculator;


    // Setup method (runs before every test)

    @Before

    public void setUp() {

        calculator = new Calculator();

        System.out.println("Setting up Calculator instance...");
```

```
}

// Teardown method (runs after every test)
@After
public void tearDown() {
    calculator = null;
    System.out.println("Cleaning up Calculator instance...");
}

@Test
public void testAdd() {
    // Arrange
    int a = 2, b = 3;

    // Act
    int result = calculator.add(a, b);

    // Assert
    assertEquals(5, result);
}

@Test
public void testSubtract() {
    int result = calculator.subtract(10, 4);
    assertEquals(6, result);
}

@Test
public void testMultiply() {
        int result = calculator.multiply(4, 3);
        assertEquals(12, result);
    }

    @Test
    public void testDivide() {
        int result = calculator.divide(20, 4);
        assertEquals(5, result);
    }

    @Test(expected = IllegalArgumentException.class)
    public void testDivideByZero() {
        calculator.divide(10, 0);
    }
}
```

Output:



```
<terminated> CalculatorTestAAA [JUnit] D:\JFSD\eclipse-jee-20
Setting up Calculator instance...
Cleaning up Calculator instance...
Setting up Calculator instance...
Cleaning up Calculator instance...
Setting up Calculator instance...
Cleaning up Calculator instance...
```

TDD using JUnit5 and Mockito

6. Mocking and Stubbing

Add Dependencies:

```
<dependency>
   <groupId>org.mockito</groupId>
   <artifactId>mockito-core</artifactId>
   <version>5.11.0</version> <!-- or latest-->
   <scope>test</scope>
</dependency>
<dependency>
   <groupId>org.junit.jupiter</groupId>
   <artifactId>junit-jupiter</artifactId>
   <version>5.10.0</version> <!-- or latest-->
   <scope>test</scope>
</dependency>
```

Created the externalAPI interface:

```
package week2;

public interface ExternalApi {
   String getData();
}
```

Created the MyService Class:

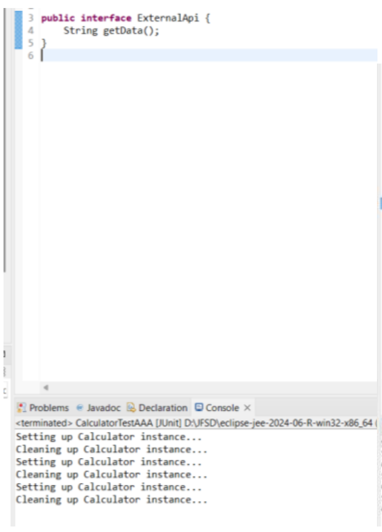
```
package week2;

public class MyService {
```

```java
    private ExternalApi externalApi;

    public MyService(ExternalApi externalApi) {
        this.externalApi = externalApi;
    }

    public String fetchData() {
        return externalApi.getData();
    }
}
```

Created the myServiceTest Class:

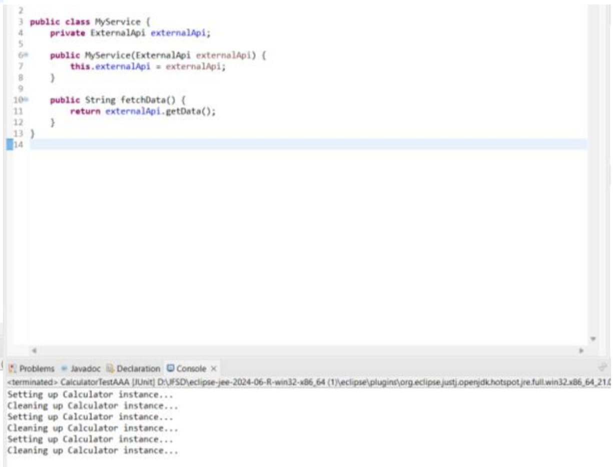
```java
package week2;

import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;
import org.mockito.Mockito;

public class MyServiceTest {

    @Test
    public void testExternalApi() {
        // Step 1: Create a mock of ExternalApi
        ExternalApi mockApi = Mockito.mock(ExternalApi.class);
        // Step 2: Stub the getData() method
        when(mockApi.getData()).thenReturn("Mock Data");

        // Step 3: Inject the mock into MyService
        MyService service = new MyService(mockApi);

        // Step 4: Call the method and assert the result
        String result = service.fetchData();
        assertEquals("Mock Data", result);
    }}
```

Output:



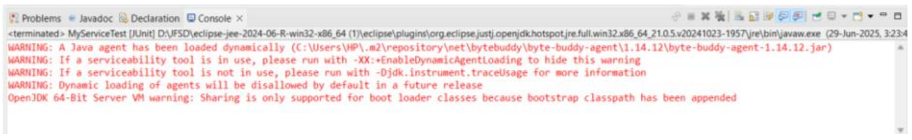
```java
public interface ExternalApi {
    String getData();
}
```

```
<terminated> CalculatorTestAAA [JUnit] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (
Setting up Calculator instance...
Cleaning up Calculator instance...
Setting up Calculator instance...
Cleaning up Calculator instance...
Setting up Calculator instance...
Cleaning up Calculator instance...
```



```java

public class MyService {
    private ExternalApi externalApi;

    public MyService(ExternalApi externalApi) {
        this.externalApi = externalApi;
    }

    public String fetchData() {
        return externalApi.getData();
    }
}
```

```
<terminated> CalculatorTestAAA [JUnit] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0
Setting up Calculator instance...
Cleaning up Calculator instance...
Setting up Calculator instance...
Cleaning up Calculator instance...
Setting up Calculator instance...
Cleaning up Calculator instance...
```



```
<terminated> MyServiceTest [JUnit] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe  (29-Jun-2025, 3:23:4
WARNING: A Java agent has been loaded dynamically (C:\Users\HP\.m2\repository\net\bytebuddy\byte-buddy-agent\1.14.12\byte-buddy-agent-1.14.12.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
```

7. Mocking and Stubbing with Multiple Returns:

Created the externalAPIStub interface:

```
package week2;

public interface ExternalApiStub {
        String getData();
}
```

Created the MyServiceStub Class:

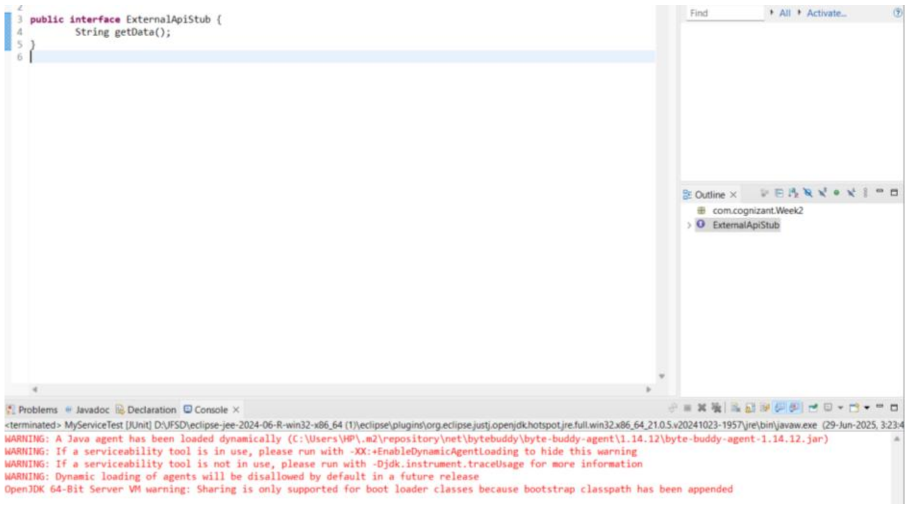
```
package week2;

public class MyServiceStub {
   private ExternalApiStub externalApiStub;

   public MyServiceStub(ExternalApiStub externalApiStub) {
      this.externalApiStub = externalApiStub;
   }

   public String fetchFirstCall() {
      return externalApiStub.getData();
   }
```

```
    public String fetchSecondCall() {
        return externalApiStub.getData();
    }
}
```

Created the MyServiceStubTest Class:

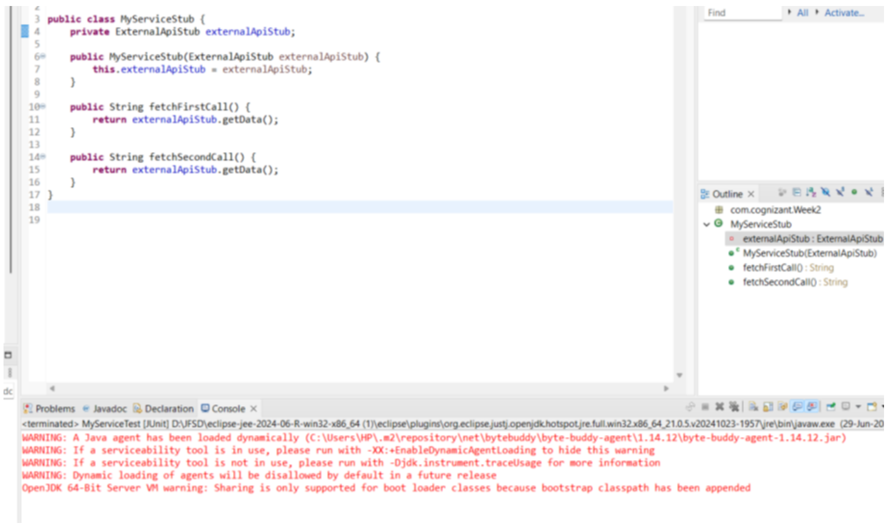
```
package week2;

import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;

public class MyServiceTestStub {

    @Test
    public void testExternalApiMultipleReturns() {
        // Step 1: Create mock object
        ExternalApiStub mockApi = mock(ExternalApiStub.class);

        // Step 2: Stub to return different values for consecutive calls
        when(mockApi.getData())
            .thenReturn("First Response")
            .thenReturn("Second Response");
        // Step 3: Inject into service
        MyServiceStub service = new MyServiceStub(mockApi);

        String first = service.fetchFirstCall();
        String second = service.fetchSecondCall();
        System.out.println("First: " + first);   // <- Print here
        System.out.println("Second: " + second);

        // Step 4: Test both calls
        assertEquals("First Response", service.fetchFirstCall());
        assertEquals("Second Response", service.fetchSecondCall());
    }
}
```

Output:



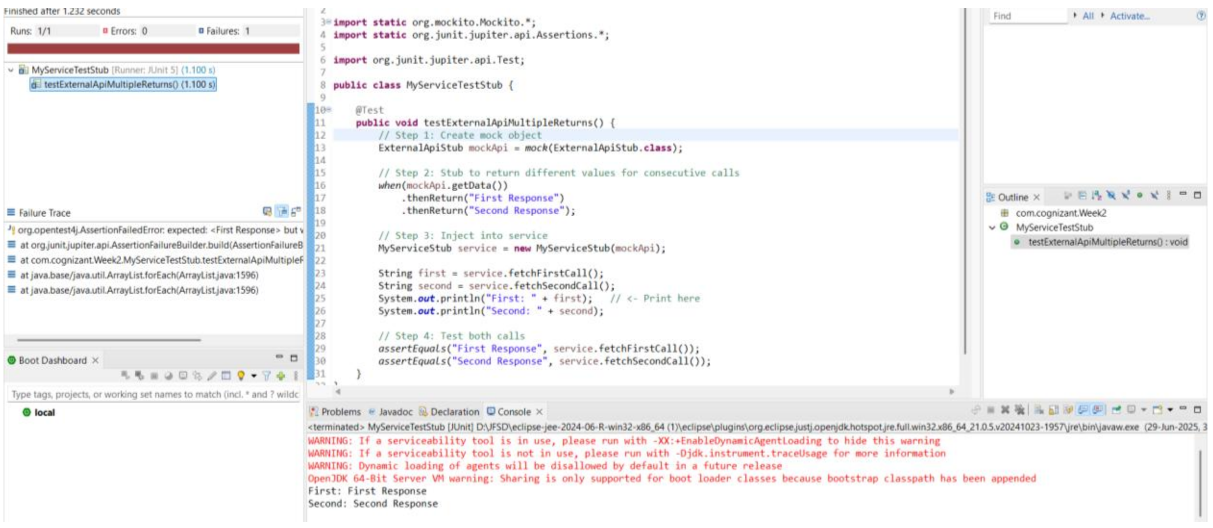
```
public interface ExternalApiStub {
        String getData();
}
```

```
<terminated> MyServiceTest [JUnit] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe (29-Jun-2025, 3:23:4
WARNING: A Java agent has been loaded dynamically (C:\Users\HP\.m2\repository\net\bytebuddy\byte-buddy-agent\1.14.12\byte-buddy-agent-1.14.12.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
```



```java
public class MyServiceStub {
    private ExternalApiStub externalApiStub;

    public MyServiceStub(ExternalApiStub externalApiStub) {
        this.externalApiStub = externalApiStub;
    }

    public String fetchFirstCall() {
        return externalApiStub.getData();
    }

    public String fetchSecondCall() {
        return externalApiStub.getData();
    }
}
```

```
<terminated> MyServiceTest [JUnit] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe (29-Jun-20
WARNING: A Java agent has been loaded dynamically (C:\Users\HP\.m2\repository\net\bytebuddy\byte-buddy-agent\1.14.12\byte-buddy-agent-1.14.12.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
```



```
Finished after 1.232 seconds
Runs: 1/1    Errors: 0    Failures: 1
MyServiceTestStub [Runner: JUnit 5] (1.100 s)
    testExternalApiMultipleReturns() (1.100 s)

Failure Trace
org.opentest4j.AssertionFailedError: expected: <First Response> but v
at org.junit.jupiter.api.AssertionFailureBuilder.build(AssertionFailureB
at com.cognizant.Week2.MyServiceTestStub.testExternalApiMultipleF
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1596)
```

```java
import static org.mockito.Mockito.*;
import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;

public class MyServiceTestStub {

    @Test
    public void testExternalApiMultipleReturns() {
        // Step 1: Create mock object
        ExternalApiStub mockApi = mock(ExternalApiStub.class);

        // Step 2: Stub to return different values for consecutive calls
        when(mockApi.getData())
            .thenReturn("First Response")
            .thenReturn("Second Response");

        // Step 3: Inject into service
        MyServiceStub service = new MyServiceStub(mockApi);

        String first = service.fetchFirstCall();
        String second = service.fetchSecondCall();
        System.out.println("First: " + first);   // <- Print here
        System.out.println("Second: " + second);

        // Step 4: Test both calls
        assertEquals("First Response", service.fetchFirstCall());
        assertEquals("Second Response", service.fetchSecondCall());
    }
```

```
<terminated> MyServiceTestStub [JUnit] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe (29-Jun-2025, 3
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
First: First Response
Second: Second Response
```

8. Verifying Interactions:

NotificationService:

```java
package week2;

public interface NotificationService {
    void sendEmail();
    void sendSMS();
    void sendPush();
}
```

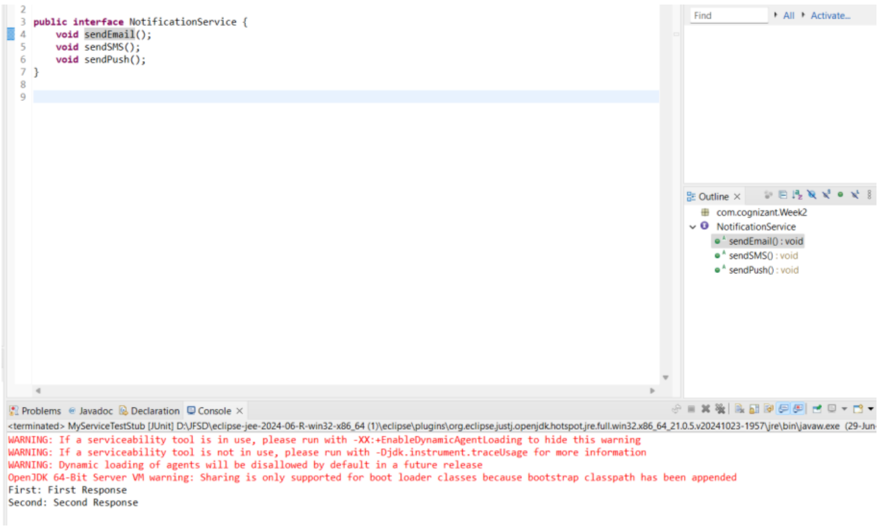
UserNotifier:

```
package week2;

public class UserNotifier {
    private NotificationService service;

    public UserNotifier(NotificationService service) {
        this.service = service;
    }

    public void notifyUser() {
        service.sendEmail();
        service.sendSMS();
        service.sendPush();
    }
}
```

UserNotifierTest:

```
package week2;

import static org.mockito.Mockito.*;
import org.junit.jupiter.api.Test;
import org.mockito.InOrder;

public class UserNotifierTest {
    @Test
    public void testNotificationOrder() {
        // Step 1: Create a mock
        NotificationService mockService = mock(NotificationService.class);

        // Step 2: Call methods in a specific order
        UserNotifier notifier = new UserNotifier(mockService);
        notifier.notifyUser();

        // Step 3: Verify interaction order
        InOrder inOrder = inOrder(mockService);
        inOrder.verify(mockService).sendEmail();
        inOrder.verify(mockService).sendSMS();
        inOrder.verify(mockService).sendPush();
    }
}
```

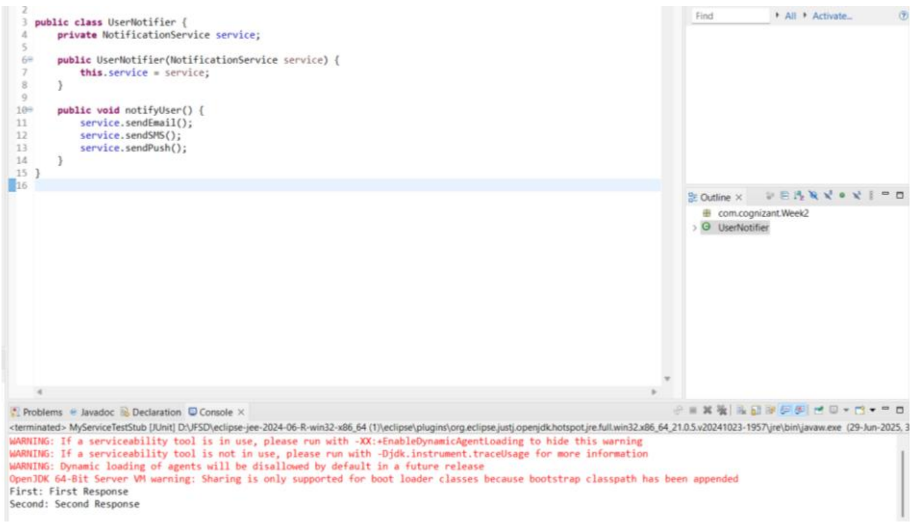
Output:



```
public interface NotificationService {
    void sendEmail();
    void sendSMS();
    void sendPush();
}
```

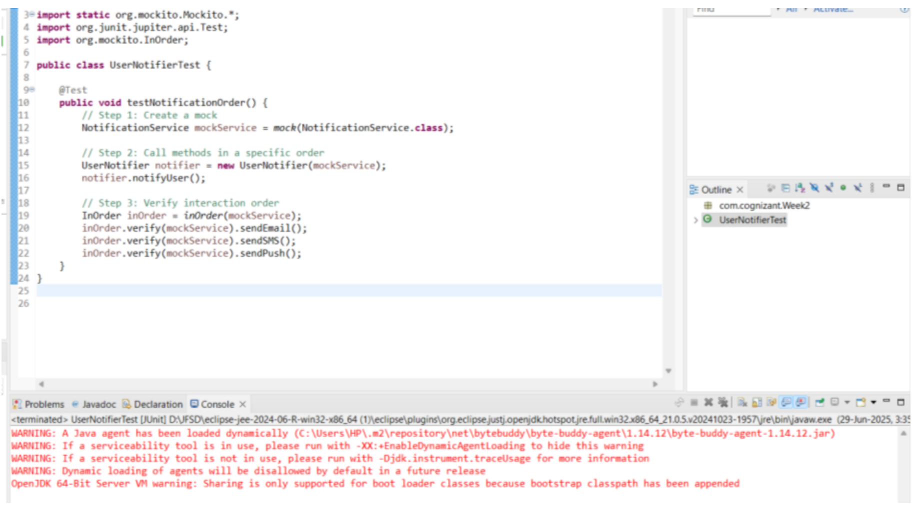
Outline: com.cognizant.Week2 / NotificationService / sendEmail() : void / sendSMS() : void / sendPush() : void

```
<terminated> MyServiceTestStub [JUnit] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe  (29-Jun
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
First: First Response
Second: Second Response
```



```java
public class UserNotifier {
    private NotificationService service;

    public UserNotifier(NotificationService service) {
        this.service = service;
    }

    public void notifyUser() {
        service.sendEmail();
        service.sendSMS();
        service.sendPush();
    }
}
```

```
<terminated> MyServiceTestStub [JUnit] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe (29-Jun-2025, 3
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
First: First Response
Second: Second Response
```



```java
import static org.mockito.Mockito.*;
import org.junit.jupiter.api.Test;
import org.mockito.InOrder;

public class UserNotifierTest {

    @Test
    public void testNotificationOrder() {
        // Step 1: Create a mock
        NotificationService mockService = mock(NotificationService.class);

        // Step 2: Call methods in a specific order
        UserNotifier notifier = new UserNotifier(mockService);
        notifier.notifyUser();

        // Step 3: Verify interaction order
        InOrder inOrder = inOrder(mockService);
        inOrder.verify(mockService).sendEmail();
        inOrder.verify(mockService).sendSMS();
        inOrder.verify(mockService).sendPush();
    }
}
```

```
<terminated> UserNotifierTest [JUnit] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe (29-Jun-2025, 3:3
WARNING: A Java agent has been loaded dynamically (C:\Users\HP\.m2\repository\net\bytebuddy\byte-buddy-agent\1.14.12\byte-buddy-agent-1.14.12.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
```

SLF4J logging framework

9. Logging Error Messages and Warning Levels:

Added Dependencies:

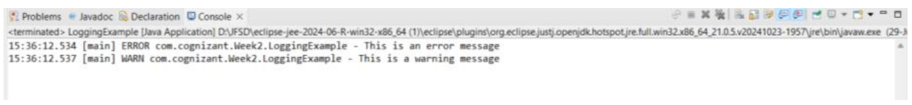
```
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.7.30</version>
```

```
</dependency>
<dependency>
   <groupId>ch.qos.logback</groupId>
   <artifactId>logback-classic</artifactId>
   <version>1.2.3</version>
</dependency>
```

Create LoggingExample.java:

```java
package week2;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
public class LoggingExample {
          private static final Logger logger = LoggerFactory.getLogger(LoggingExample.class);

            public static void main(String[] args) {
               logger.error("This is an error message");
               logger.warn("This is a warning message");
            }
}
```

Output:



```
<terminated> LoggingExample [Java Application] D:\JFSD\eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe  (29-
15:36:12.534 [main] ERROR com.cognizant.Week2.LoggingExample - This is an error message
15:36:12.537 [main] WARN com.cognizant.Week2.LoggingExample - This is a warning message
```