Week 2 Assignment

**Module 2 : Pl/SQL Programming**

**Exercise 1 : Control Structures**

**Scenario 1 : Discount for Customers Above 60**

**DECLARE**

**CURSOR cust\_cursor IS**

**SELECT customer\_id, age, loan\_interest\_rate**

**FROM customers;**

**BEGIN**

**FOR cust\_rec IN cust\_cursor LOOP**

**IF cust\_rec.age > 60 THEN**

**UPDATE customers**

**SET loan\_interest\_rate = loan\_interest\_rate \* 0.99**

**WHERE customer\_id = cust\_rec.customer\_id;**

**END IF;**

**END LOOP;**

**COMMIT;**

**END;**

**/**

**Scenario 2 : Promote to VIP Based on Balance**

**DECLARE**

**CURSOR vip\_cursor IS**

**SELECT customer\_id, balance**

**FROM customers;**

**BEGIN**

**FOR vip\_rec IN vip\_cursor LOOP**

**IF vip\_rec.balance > 10000 THEN**

**UPDATE customers**

**SET isvip = 'TRUE'**

**WHERE customer\_id = vip\_rec.customer\_id;**

**END IF;**

**END LOOP;**

**COMMIT;**

**END;**

**/**

**Scenario 3: Loan Reminders for Due Dates Within 30 Days**

**DECLARE**

**CURSOR loan\_cursor IS**

**SELECT loan\_id, customer\_id, due\_date**

**FROM loans**

**WHERE due\_date BETWEEN SYSDATE AND SYSDATE + 30;**

**BEGIN**

**FOR loan\_rec IN loan\_cursor LOOP**

**DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || loan\_rec.loan\_id ||**

**' for Customer ' || loan\_rec.customer\_id ||**

**' is due on ' || TO\_CHAR(loan\_rec.due\_date, 'DD-MON-YYYY'));**

**END LOOP;**

**END;**

**/**

**Exercise 3 : Stored Procedures**

**Scenario 1 : Process Monthly Interest**

**CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS**

**BEGIN**

**UPDATE savings\_accounts**

**SET balance = balance \* 1.01;**

**COMMIT;**

**END;**

**/**

**Scenario 2 : Update Employee Bonus**

**CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(**

**p\_department\_id IN NUMBER,**

**p\_bonus\_percent IN NUMBER) IS**

**BEGIN**

**UPDATE employees**

**SET salary = salary + (salary \* p\_bonus\_percent / 100)**

**WHERE department\_id = p\_department\_id;**

**COMMIT;**

**END;**

**/**

**Scenario 3: Transfer Funds Between Accounts**

**CREATE OR REPLACE PROCEDURE TransferFunds(**

**p\_from\_account\_id IN NUMBER,**

**p\_to\_account\_id IN NUMBER,**

**p\_amount IN NUMBER) IS**

**v\_balance NUMBER;**

**BEGIN**

**SELECT balance INTO v\_balance**

**FROM accounts**

**WHERE account\_id = p\_from\_account\_id;**

**IF v\_balance < p\_amount THEN**

**RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance');**

**END IF;**

**UPDATE accounts**

**SET balance = balance - p\_amount**

**WHERE account\_id = p\_from\_account\_id;**

**UPDATE accounts**

**SET balance = balance + p\_amount**

**WHERE account\_id = p\_to\_account\_id;**

**COMMIT;**

**END;**

**/**

**Module 3 : Junit Testing**

**Exercise 1 : Setting up Junit**

**CODE:**

**Calculator.java:**

**package example;**

**public class Calculator {**

**public int add(int a, int b) {**

**return a + b;**

**}**

**public int subtract(int a, int b) {**

**return a - b;**

**}**

**}**

**CalculatorTest.java:**

**package example;**

**import org.junit.Test;**

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

**public class CalculatorTest {**

**@Test**

**public void testAdd() {**

**Calculator calc = new Calculator();**

**int result = calc.add(5,3);**

**assertEquals(8, result);**

**}**

**}**

**Pom.xml:**

**<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.example</groupId>**

**<artifactId>settingupJUnit</artifactId>**

**<version>1.0-SNAPSHOT</version>**

**<name>settingupJUnit</name>**

**<description>A project for setting up JUnit</description>**

**<dependencies>**

**<dependency>**

**<groupId>junit</groupId>**

**<artifactId>junit</artifactId>**

**<version>4.13.2</version>**

**<scope>test</scope>**

**</dependency>**

**</dependencies>**

**<build>**

**<plugins>**

**<plugin>**

**<groupId>org.apache.maven.plugins</groupId>**

**<artifactId>maven-surefire-plugin</artifactId>**

**<version>3.0.0</version>**

**</plugin>**

**</plugins>**

**</build>**

**</project>**

**Exercise 3 : Assertions in Junit**

**CODE:**

**AssertionsTest.java:**

**package com.example;**

**import org.junit.Test;**

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

**public class AssertionsTest {**

**@Test**

**public void testAssertions() {**

**assertEquals(5, 2 + 3);**

**assertTrue(5 > 3);**

**assertFalse(5 < 3);**

**assertNull(null);**

**assertNotNull(new Object());**

**}**

**}**

**Pom.xml:**

**<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.example</groupId>**

**<artifactId>JUnitDemo</artifactId>**

**<version>1.0-SNAPSHOT</version>**

**<dependencies>**

**<!-- JUnit 4 -->**

**<dependency>**

**<groupId>junit</groupId>**

**<artifactId>junit</artifactId>**

**<version>4.13.2</version>**

**<scope>test</scope>**

**</dependency>**

**</dependencies>**

**</project>**

**Exercise 4:** **Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in Junit**

**CODE:**

**CalculatorFixtureTest.java**

**<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.example</groupId>**

**<artifactId>JunitDemo</artifactId>**

**<version>1.0-SNAPSHOT</version>**

**<dependencies>**

**<!—JUnit 4**

**<dependency>**

**<groupId>junit</groupId>**

**<artifactId>junit</artifactId>**

**<version>4.13.2</version>**

**<scope>test</scope>**

**</dependency>**

**</dependencies>**

**</project>**

**Module 3 :Mockito Exercise**

**Exercise 1: Mocking and Stubbing**

**CODE:**

**ExternalApi:**

**package com.example;**

**public interface ExternalApi {**

**String getData();**

**}**

**MyService:**

**package com.example;**

**public class MyService {**

**private final ExternalApi api;**

**public MyService(ExternalApi api) {**

**this.api = api;**

**}**

**public String fetchData() {**

**return api.getData();**

**}**

**}**

**MyServiceTest:**

**package com.example;**

**import org.junit.jupiter.api.Test;**

**import org.mockito.Mockito;**

**import static org.mockito.Mockito.when;**

**import static org.junit.jupiter.api.Assertions.assertEquals;**

**public class MyServiceTest {**

**@Test**

**public void testExternalApi() {**

**ExternalApi mockApi = Mockito.mock(ExternalApi.class);**

**when(mockApi.getData()).thenReturn("Mock Data");**

**MyService service = new MyService(mockApi);**

**String result = service.fetchData();**

**assertEquals("Mock Data", result);**

**}**

**}**

**Pom.xml:**

**<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.example</groupId>**

**<artifactId>MockingandStubbing</artifactId>**

**<version>1.0-SNAPSHOT</version>**

**<properties>**

**<maven.compiler.source>17</maven.compiler.source>**

**<maven.compiler.target>17</maven.compiler.target>**

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

**</properties>**

**<dependencies>**

**<!-- JUnit 5 -->**

**<dependency>**

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

**<artifactId>junit-jupiter</artifactId>**

**<version>5.9.1</version>**

**<scope>test</scope>**

**</dependency>**

**<!-- Mockito -->**

**<dependency>**

**<groupId>org.mockito</groupId>**

**<artifactId>mockito-core</artifactId>**

**<version>4.8.0</version>**

**<scope>test</scope>**

**</dependency>**

**</dependencies>**

**<build>**

**<plugins>**

**<!-- Ensure JUnit 5 runs correctly -->**

**<plugin>**

**<groupId>org.apache.maven.plugins</groupId>**

**<artifactId>maven-surefire-plugin</artifactId>**

**<version>2.22.2</version>**

**<configuration>**

**<useModulePath>false</useModulePath>**

**</configuration>**

**</plugin>**

**<!-- Set compiler to Java 17 -->**

**<plugin>**

**<groupId>org.apache.maven.plugins</groupId>**

**<artifactId>maven-compiler-plugin</artifactId>**

**<version>3.11.0</version>**

**<configuration>**

**<source>17</source>**

**<target>17</target>**

**</configuration>**

**</plugin>**

**</plugins>**

**</build>**

**</project>**

**Exercise 2: Verifying Interactions**

**CODE:**

**ExternalApi:**

**package com.example;**

**public interface ExternalApi {**

**void getData();**

**}**

**MyService.java:**

**package com.example;**

**public class MyService {**

**private final ExternalApi api;**

**public MyService(ExternalApi api) {**

**this.api = api;**

**}**

**public void fetchData() {**

**api.getData();**

**}**

**}**

**MyServiceTest.java:**

**package com.example;**

**import org.junit.jupiter.api.Test;**

**import static org.mockito.Mockito.\*;**

**public class MyServiceTest {**

**@Test**

**public void testVerifyInteraction() {**

**ExternalApi mockApi = mock(ExternalApi.class);**

**MyService service = new MyService(mockApi);**

**service.fetchData();**

**verify(mockApi).getData(); // Verifies it was called once**

**}**

**}**

**Pom.xml:**

**<project>**

**<modelVersion>4.0.0</modelVersion>**

**<groupId>com.example</groupId>**

**<artifactId>MockingAndStubbing</artifactId>**

**<version>1.0-SNAPSHOT</version>**

**<dependencies>**

**<!-- JUnit 5 -->**

**<dependency>**

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

**<artifactId>junit-jupiter</artifactId>**

**<version>5.10.0</version>**

**<scope>test</scope>**

**</dependency>**

**<!-- Mockito -->**

**<dependency>**

**<groupId>org.mockito</groupId>**

**<artifactId>mockito-core</artifactId>**

**<version>5.11.0</version>**

**<scope>test</scope>**

**</dependency>**

**</dependencies>**

**<build>**

**<plugins>**

**<!-- Set Java version -->**

**<plugin>**

**<groupId>org.apache.maven.plugins</groupId>**

**<artifactId>maven-compiler-plugin</artifactId>**

**<version>3.11.0</version>**

**<configuration>**

**<source>17</source>**

**<target>17</target>**

**</configuration>**

**</plugin>**

**</plugins>**

**</build>**

**</project>**

**Module 4 : SL4J Logging**

**Exercise 1: Logging Error Messages and Warning Levels**

**CODE:**

**LoggingExample.java:**

**package com.example;**

**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");**

**logger.info("This is an info message");**

**logger.debug("This is a debug message");**

**logger.trace("This is a trace message");**

**}**

**}**

**Pom.xml:**

**<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.example</groupId>**

**<artifactId>SLF4JLoggingExample</artifactId>**

**<version>1.0-SNAPSHOT</version>**

**<dependencies>**

**<!-- SLF4J API -->**

**<dependency>**

**<groupId>org.slf4j</groupId>**

**<artifactId>slf4j-api</artifactId>**

**<version>1.7.30</version>**

**</dependency>**

**<!-- Logback Classic (SLF4J binding) -->**

**<dependency>**

**<groupId>ch.qos.logback</groupId>**

**<artifactId>logback-classic</artifactId>**

**<version>1.2.3</version>**

**</dependency>**

**</dependencies>**

**</project>**

**Logback.xml:**

**<configuration>**

**<appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">**

**<encoder>**

**<pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>**

**</encoder>**

**</appender>**

**<root level="debug">**

**<appender-ref ref="STDOUT"/>**

**</root>**

**</configuration>**