

https://drive.google.com/file/d/15SJV_EeTilLiGUpd7K4efw0aLDcBE9TQ/view?usp=sharing

```
<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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>my.unit.test</groupId>
<artifactId>unit-test-assignment</artifactId>
<version>0.0.1-SNAPSHOT</version>
<properties>
<java.version>17</java.version>
<project.build.sourceEncoding>utf-8</project.build.sourceEncoding>
</properties>
<dependencies>
<dependency>
<groupId>com.google.guava</groupId>
<artifactId>guava</artifactId>
<version>30.1.1-jre</version>
</dependency>
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter</artifactId>
<version>5.7.2</version>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.assertj</groupId>
<artifactId>assertj-core</artifactId>
<version>3.20.2</version>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-junit-jupiter</artifactId>
<version>3.11.2</version>
<scope>test</scope>
</dependency>
</dependencies>
<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.8.1</version>
<configuration>
```

```

<source>${java.version}</source>
<target>${java.version}</target>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

```

package com.promineotech;
import java.util.Random;
public class TestDemo {
public int addPositive(int a, int b) {
    if(a>=0 && b>=0) {
        int sum = a + b;
        System.out.println(sum);
        return sum;
    } else {
        throw new IllegalArgumentException("Both parameters must be positive!");
    }
}

int getRandomInt() {
    Random random = new Random();
    return random.nextInt(10) + 1;
}

public int randomNumberSquared() {
    int randomNumber = getRandomInt();
    int randomNumberSquared = randomNumber * randomNumber;
    return randomNumberSquared;
}
}

```

```

package com.promineotech;
import static org.assertj.core.api.Assertions.assertThat;
import static org.assertj.core.api.Assertions.assertThatThrownBy;
import static org.mockito.Mockito.doReturn;
import static org.mockito.Mockito.spy;
import java.util.stream.Stream;
import org.junit.jupiter.params.provider.Arguments;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Disabled;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.MethodSource;
class TestDemoJUnitTest {
    TestDemo testDemo;
    @BeforeEach

```

```

    void setUp() throws Exception {
        testDemo = new TestDemo();
    }
    @ParameterizedTest
    @MethodSource("com.promineotech.TestDemoJUnitTest#argumentsForAddPositive")
    void assertThatTwoPositiveNumbersAreAddedCorrectly(int a, int b, int expected, boolean
exceptException) {
        if (!exceptException) {
            assertThat(testDemo.addPositive(a, b)).isEqualTo(expected);
        } else
            assertThatThrownBy(() -> testDemo.addPositive(a,
b)).isInstanceOf(IllegalArgumentException.class);
        }
        static Stream<Arguments> argumentsForAddPositive() {
            return Stream.of(Arguments.of(1, 2, 3, false),
                Arguments.of(2, 4, 6, false),
                Arguments.of(3, 6, 9, false),
                Arguments.of(2, -4, 6, true),
                Arguments.of(0, 3, 3, false)
                // , Arguments.of(-7, 4, 2, false)
            );
        }

        @Test
        @Disabled
        void assertThatPairsOfPositiveNumbersAreAddedCorrectly() {
            TestDemo mockDemo = spy(testDemo);

            doReturn(5).when(mockDemo).getRandomInt();

            int fiveSquared = mockDemo.randomNumberSquared();

            assertThat(fiveSquared).isEqualTo(25);

            assertThat(testDemo.addPositive(4,5)).isEqualTo(9);

            assertThat(testDemo.addPositive(40,50)).isEqualTo(90);

        }
    }
}

```