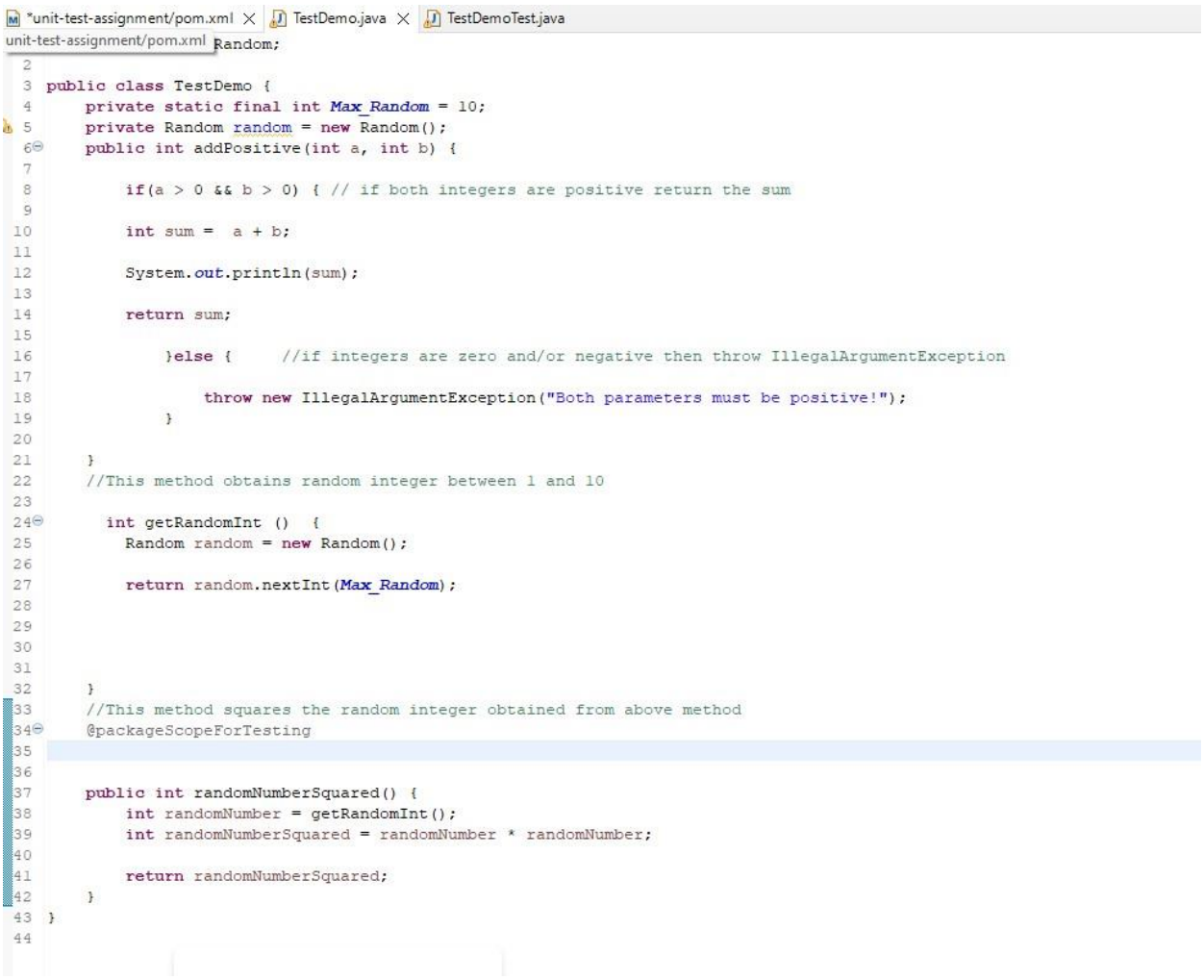
https://github.com/bmason1969/Week-12.git

*unit-test-assignment/pom.xml | TestDemo.java | TestDemoTest.java

```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 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>5.9.0</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.assertj</groupId>
            <artifactId>assertj-core</artifactId>
            <version>3.23.1</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.mockito</groupId>
            <artifactId>mockito-junit-jupiter</artifactId>
            <version>4.8.0</version>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
```

```xml
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.10.1</version>
                <configuration>
                    <source>${java.version}</source>
                    <target>${java.version}</target>
                </configuration>
            </plugin>
        </plugins>
    </build>


</project>
```


*unit-test-assignment/pom.xml
TestDemo.java
TestDemoTest.java
unit-test-assignment/pom.xml
Random;
public class TestDemo {
private static final int Max_Random = 10;
private Random random = new Random();
public int addPositive(int a, int b) {
if(a > 0 && b > 0) { // if both integers are positive return the sum
int sum = a + b;
System.out.println(sum);
return sum;
}else { //if integers are zero and/or negative then throw IllegalArgumentException
throw new IllegalArgumentException("Both parameters must be positive!");
}
}
//This method obtains random integer between 1 and 10
int getRandomInt () {
Random random = new Random();
return random.nextInt(Max_Random);
}
//This method squares the random integer obtained from above method
@packageScopeForTesting
public int randomNumberSquared() {
int randomNumber = getRandomInt();
int randomNumberSquared = randomNumber * randomNumber;
return randomNumberSquared;
}
}

*unit-test-assignment/pom.xml | TestDemo.java | *TestDemoTest.java

```
22      private TestDemo testDemo;
23
24      @BeforeEach
25      void setUp() throws Exception {
26          testDemo = new TestDemo();
27      }
28
29
30      @ParameterizedTest
31      @MethodSource("TestDemoTest#argumentsForAddPositive#")
32      void assertThatTwoPositiveNumbersAreAddedCorrectly(int a, int b, int expected,
33          boolean expectException){
34
35
36
37          if (!expectException) {
38              assertThatThrownBy(() ->
39      testDemo.addPositive(a, b))
40          .isInstanceOf(IllegalArgumentException.class);
41          }
42
43
44          else {
45              int actual = testDemo.addPositive(a, b);
46          assertThat(actual).isEqualTo(expected);
47      }
48  }
49      @Test
50
51      void assertThatNumberSquaredIsCorrect () {
52      int expected = 25;
53      int a = 5;
54
55
56
57      TestDemo mockDemo = spy(testDemo);
58      doReturn(5).when(mockDemo).getRandomInt();
59      int actual =mockDemo.randomNumberSquared();
60      assertThat(actual).isEqualTo(expected);
61  }
62
63
64
65      static Stream<Arguments> argumentsForAddPositive() {
66          //@formatter: off
67
```

```
67
68      return Stream.of(
69              arguments(5, 5, 25, true),
70              arguments (5, 1, 6, true),
71              arguments(23,0, 23, true),
72              arguments(-4, 2, -6, false)
73              );
74
75      //@formatter: on
76      }
77
78  }
79
80
81
82
```