

URL to GitHub Repository: <https://git@github.com/benn51/week12JUnitTesting.git>

URL to Public Link of your Video: https://youtu.be/8_47AJ1qrKg

TestDemo Class

```
import java.util.Random;
```

```
public class TestDemo {
```

```
    public int addPositive(int a, int b) {
```

```
        if(a>0 && b>0) {
```

```
            return a+b;
```

```
        }
```

```
    else {
```

```
        throw new IllegalArgumentException("Both Parameters need to be positive");
```

```
    }
```

```
}
```

```
int randomNumberSquared() {
```

```
    int ranadomNumber =getRandomInt();
```

```
    int result= ranadomNumber * ranadomNumber;
```

```
    return result;
```

```
}

int getRandomInt() {

    Random random = new Random();

    return random.nextInt(10) + 1;

}

}
```

TestDemoTest Class

```
import java.util.stream.Stream;
```

```
import static org.assertj.core.api.Assertions.assertThat,

import static org.junit.jupiter.params.provider.Arguments.arguments;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.params.ParameterizedTest;

import org.junit.jupiter.params.provider.Arguments;

import org.junit.jupiter.params.provider.MethodSource;

import static org.assertj.core.api.Assertions.assertThatThrownBy,

import static org.mockito.Mockito.doReturn,

import static org.mockito.Mockito.spy,

class TestDemoTest {

    private TestDemo testDemo;

    @BeforeEach

    void setUp() {

        testDemo = new TestDemo();

    }

    // command + shift + O should resolve all imports on project in Eclipse
```

// Test arguments with values to determine if expectException is thrown

```
static Stream<Arguments> argumentsForAddPositive() {  
  
    return(Stream.of(arguments(1, 2, 3, false), arguments(4, 0, 4, true),  
  
arguments(3, -4, -1, true), arguments(-4, -5, -9, true),  
  
arguments(0, 0, 0, true), arguments(5, 3, 8, false),  
  
arguments(55, 55, 110, false), arguments(-9, 2, -7, true),  
  
arguments(13, 200, 213, false), arguments(999, 1, 1000, false)  
  
));  
  
}
```

@ParameterizedTest

@MethodSource("TestDemoTest#argumentsForAddPositive")

```
void assertThatTwoPositiveNumbersAreAddedCorrectly(int a, int b, int expected,  
Boolean expectException) {
```

//Given: two positive integers

```
if(!expectException) {
```

```
    assertThat(testDemo.addPositive(a, b)).isEqualTo(expected);
```

```
} else {
```

```
    assertThatThrownBy(() -> testDemo.addPositive(a, b))
```

```
.assertInstanceOf(IllegalArgumentException.class);
```

```
}
```

```
//When: the method to add positive integers together is called
```

```
//Then: sum of positive integers added together is returned
```

```
}
```

```
@Test
```

```
void assertThatNumberSquaredIsCorrect() {
```

```
    TestDemo mockDemo = spy(testDemo);
```

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

```
    int fiveSquared = mockDemo.randomNumberSquared();
```

```
    assertThat(fiveSquared).isEqualTo(25);
```

```
}
```

```
}
```