

# @BeforeAll and @AfterAll Annotation

This lesson demonstrates working of two more Lifecycle methods annotated with - @BeforeAll and @AfterAll Annotation.

## We'll cover the following ^

- @BeforeAll and @AfterAll

## @BeforeAll and @AfterAll #

Methods annotated with `@BeforeAll` and `@AfterAll` are `static` methods because, as the name suggest they are called once before all and once after all `@Test` methods. So, if in a test class there are two `@Test` methods, the `@BeforeAll` method will be called once before all test methods and similarly, the `@AfterAll` method will be called once, just after all `@Test` method gets finished. Let's look at demo taking previous lesson test class:-

```
1 package io.educative.junit5;
2
3 import org.junit.jupiter.api.A
4 import org.junit.jupiter.api.A
5 import org.junit.jupiter.api.Be
6 import org.junit.jupiter.api.Be
7 import org.junit.jupiter.api.Te
8
9 public class LifecycleTest {
10
11     @BeforeAll
12     public static void beforeAll() {
13         System.out.println("Li
14     }
15
16     public LifecycleTest() {
17         System.out.println("Li
18     }
19
20     @BeforeEach
21     public void beforeEach() {
22         System.out.println("Li
23     }
24
25     @Test
26     public void testOne() {
```



```

27     System.out.println("Li
28     }
29
30     @Test
31     public void testTwo() {

```



```

<terminated> LifecycleTest [JUnit] /Library/Java/JavaVirtualMachines/jdk1.8.0_111.jdk/Contents/Home/bin/ja
LifecycleTest - beforeAll() got executed !!!
LifecycleTest - Constructor got executed !!!
LifecycleTest - beforeEach() got executed !!!
LifecycleTest - testOne() got executed !!!
LifecycleTest - afterEach() got executed !!!
LifecycleTest - Constructor got executed !!!
LifecycleTest - beforeEach() got executed !!!
LifecycleTest - testTwo() got executed !!!
LifecycleTest - afterEach() got executed !!!
LifecycleTest - afterAll() got executed !!!

```

Output of the test

You can see in the output of the above test class, that `@BeforeAll` and `@AfterAll` has the following properties:-

1. Both are static methods.
2. Both are called once in a test lifecycle.
3. `@BeforeAll` annotated method being a class level method, it gets called even before the constructor.
4. `@AfterAll` annotated method being a class level method, it gets called after all methods get executed.
5. `@BeforeAll` annotated method is used where expensive resource initialization needs to be done such as database connection, server start etc. This resources that can be used by test methods.
6. `@AfterAll` annotated method is used where expensive resource clean up needs to be done such as, database connection terminates, server stop etc. So that these expensive resources can be cleaned up.

In the next chapter, we will discuss about `@DisplayName`.

