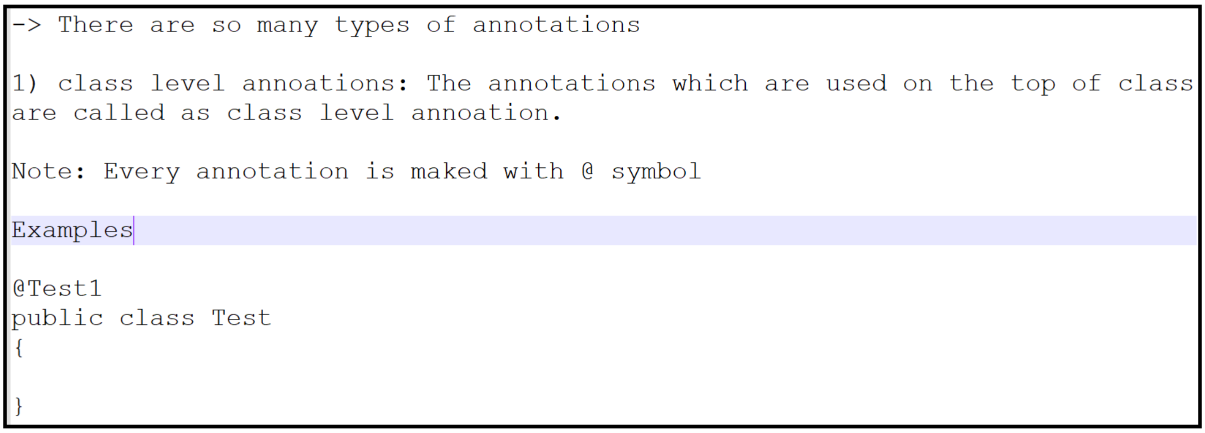


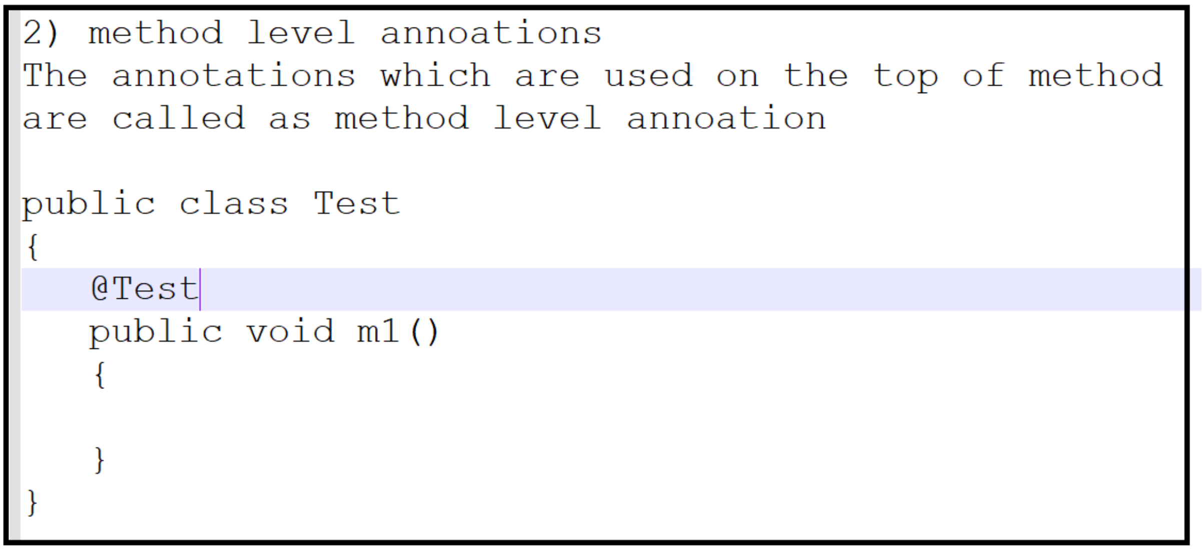
-> There are so many types of annotations

1) class level annoations: The annotations which are used on the top of class are called as class level annoation.

Note: Every annotation is maked with @ symbol

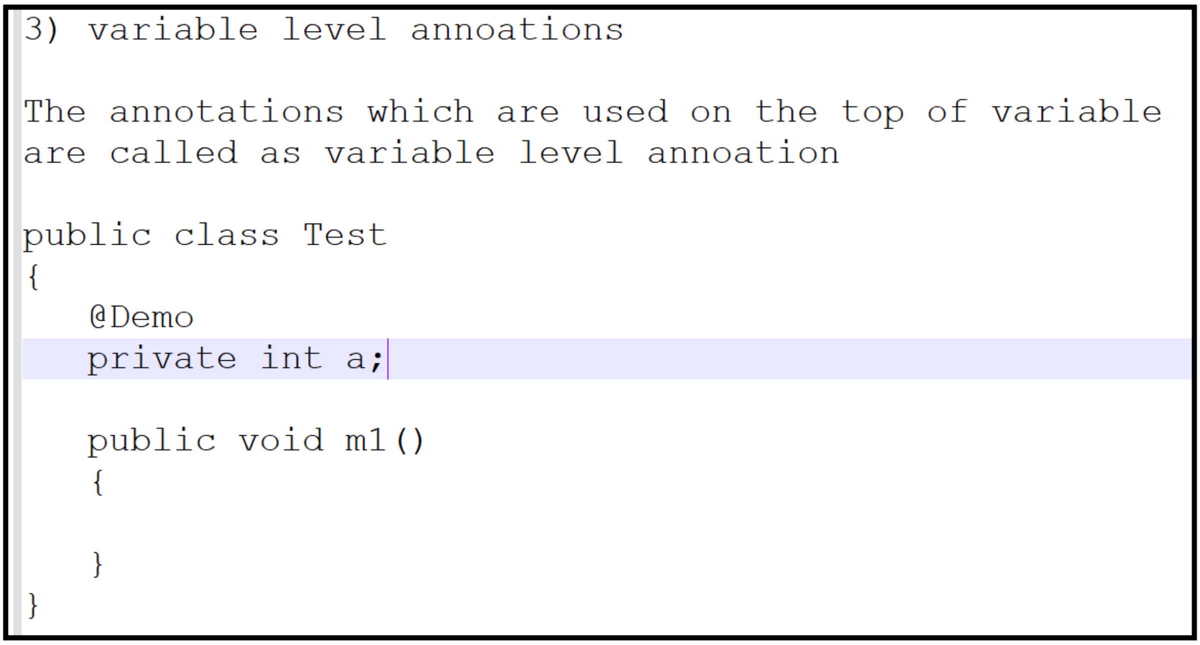
Examples

```
@Test1
public class Test
{

}
```



2) method level annoations
The annotations which are used on the top of method are called as method level annoation

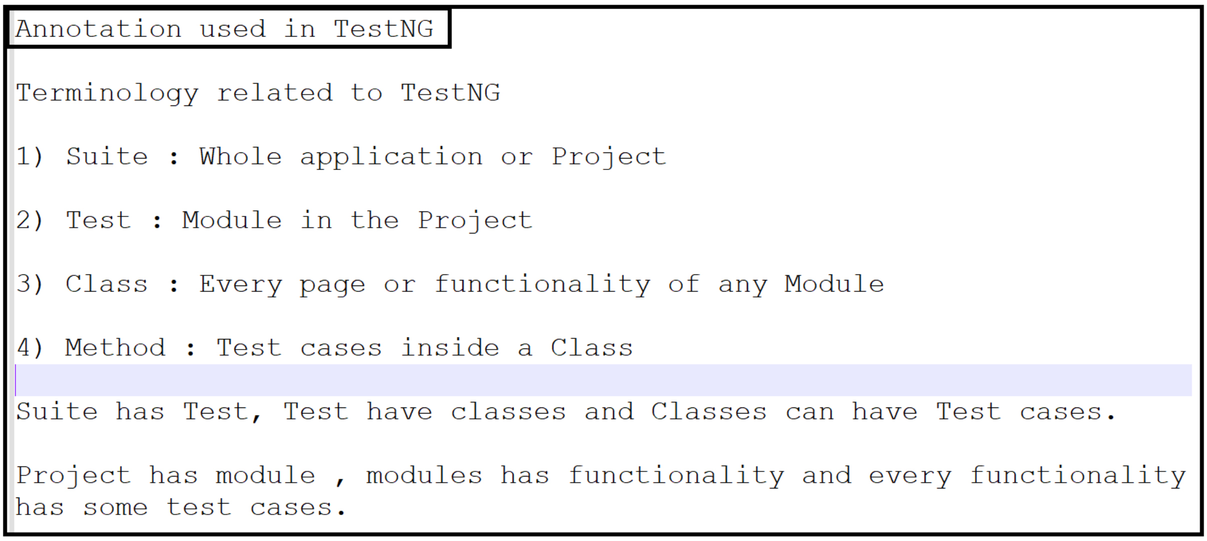
```
public class Test
{
    @Test
    public void m1()
    {

    }
}
```



3) variable level annoations

The annotations which are used on the top of variable are called as variable level annoation

```
public class Test
{
    @Demo
    private int a;

    public void m1()
    {

    }
}
```



Annotation used in TestNG

Terminology related to TestNG

1) Suite : Whole application or Project

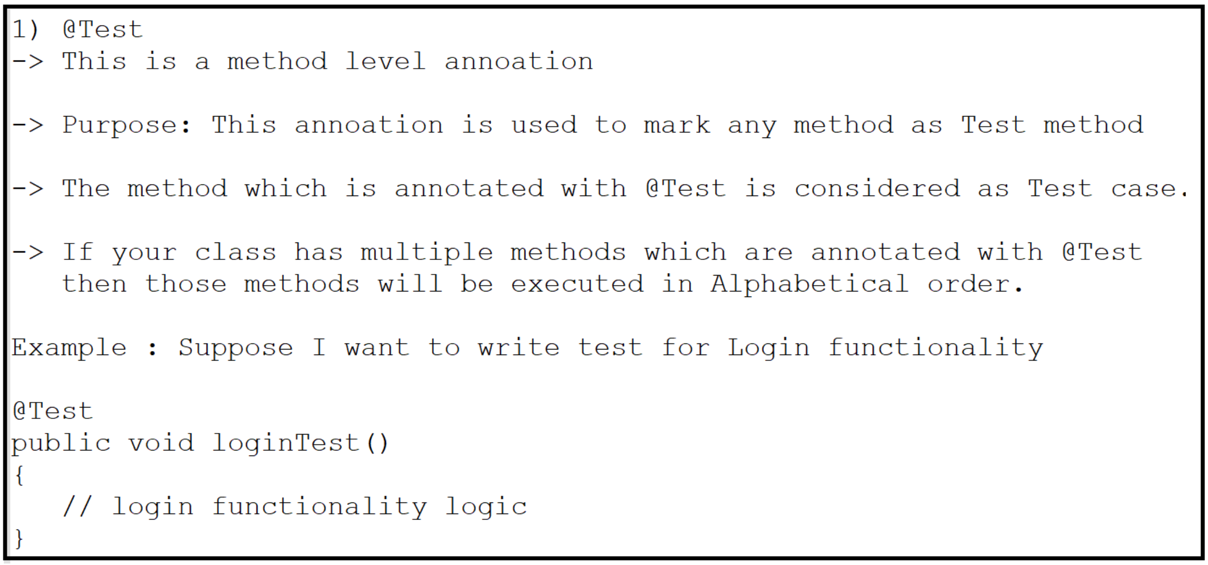
2) Test : Module in the Project

3) Class : Every page or functionality of any Module

4) Method : Test cases inside a Class

Suite has Test, Test have classes and Classes can have Test cases.

Project has module , modules has functionality and every functionality has some test cases.



1) @Test
-> This is a method level annoation

-> Purpose: This annoation is used to mark any method as Test method

-> The method which is annotated with @Test is considered as Test case.

-> If your class has multiple methods which are annotated with @Test
   then those methods will be executed in Alphabetical order.

Example : Suppose I want to write test for Login functionality

```
@Test
public void loginTest()
{
   // login functionality logic
}
```

2) @BeforeTest

-> This is method level annoation

-> The method which is annotated with @BeforeTest get executed before all @Test annotated method

-> If there are multiple methods with @BeforeTest annoation then those methods get executed in Alphabetical order

-> When we should go for @BeforeTest annoation?

Ans: Before executing every test case some times we need to peforme some clean up activity in that case, we write that clean activity inside @BeforeTest annotated method

Before executing every test case some we need to set some common data, in that case we will add that code inside @BeforeTest annotated method

Example : Object Creation logic, File opening logic,
DB Connection logic ,Browser Opening logic

-> The method annoated with @BeforeTest is executed only one time in life cycle

3) @AfterTest

-> It is a method level annoation

-> The method which is annoated with @AfterTest get executed after executing all test cases.

-> If there are multiple methods with @AfterTest annotation then those methods get executed in Alphabetical order

-> @AfterTest annotated method get executed only ones

-> When we should go for @AfterTest annoation?

Ans: After executing all test cases, if we want to peforme some clean up activity then that clean up activites code we need to write inside @AfterTest annoated method

Example :
Browser closing logic
db connection logic
File closing logic

4) @BeforeMethod

-> This is method level annoation

-> The method annotated with @BeforMethod get executed before every test case execution but get executed after the @BeforeTest annotated method.

-> When we should go for @BeforeMethod annoation?
Ans:If we want to peform some common prerequisite for every test case then we will write that code inside @BeforeMethod annotated method so that we no need to write same code again and again

5) @AfterMethod

-> It is method level annotation

-> The method which is annotated with @AfterMethod get executed after every test case execution

-> When we should go for @AfterMethod annoation?
Ans: after executing every teat case if we want to perform some common operation then we should go for @AfterMethod annotation

6) @BeforeClass

-> This is method level annotation

-> The method which is annotated with @BeforeClass get executed before execution of @BeforeTest annotated method in the current class.

-> This method get executed only one time

-> If there are multiple methods with @BeforeClass annotation then all methods get executed in Alphabetical order

-> When we should go for @BeforeClass annotation?
Ans : Before executing every Test case of current class if we want to perform some common operation then that code we need to write inside @BeforeClass annotated method.

7) @AfterClass

-> It is method level annoation

-> The method which is annotated with @AfterClass get executed after executing all methods which are annotated with @AfterMethod

8) @BeforeSuite

-> It is method level annoation

-> The method which is annotated with @BeforeSuite get executed before all tests in this suite.

-> This annotation has priority 1

-> The method which is annoated with @BeforeSuite get executed only ones.

-> If there are multiple methods with @BeforeSuite annoation then those methods get executed in Alphabetical order

-> When we should go for @BeforeSuite annoation?

Ans: Some common data which is required in every module that common data setting logic we can write inside @BeforeSuite annotated method

Example : Suppose Employee.xml file is required in every module that file loading logic we will write inside @BeforeSuite annoated methods

@AfterSuite

-> It is method level annoation

-> The method which is annotated with @AfterSuite get executed after all tests get executed in this suite.

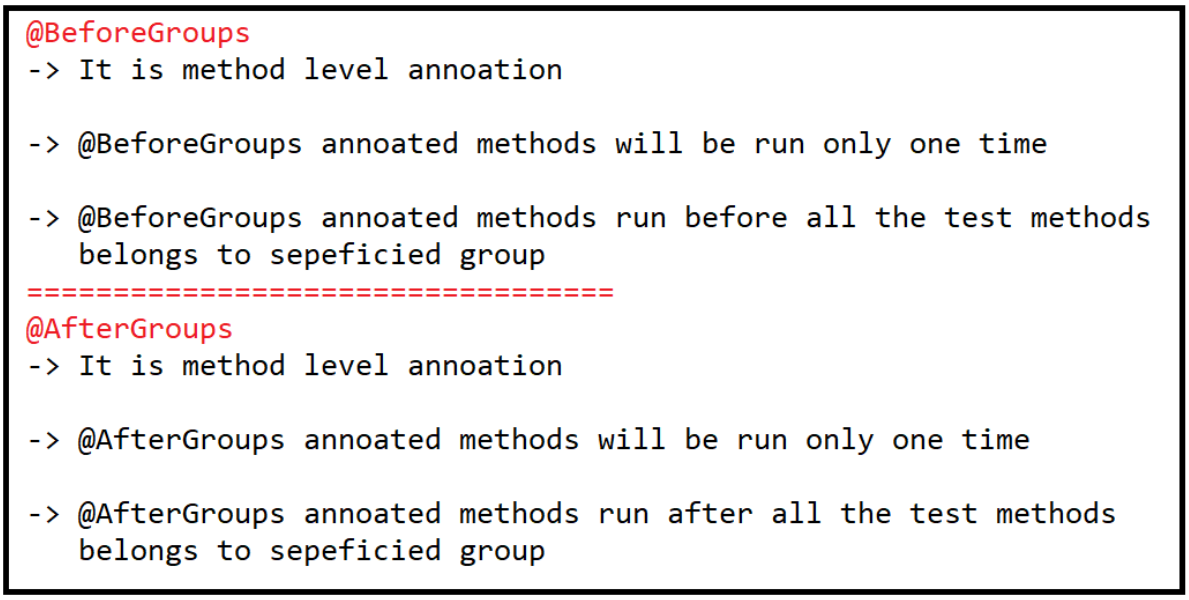
-> It has lowest priority among the TestNg annoation

-> The method which is annoated with @AfterSuite get executed only ones.

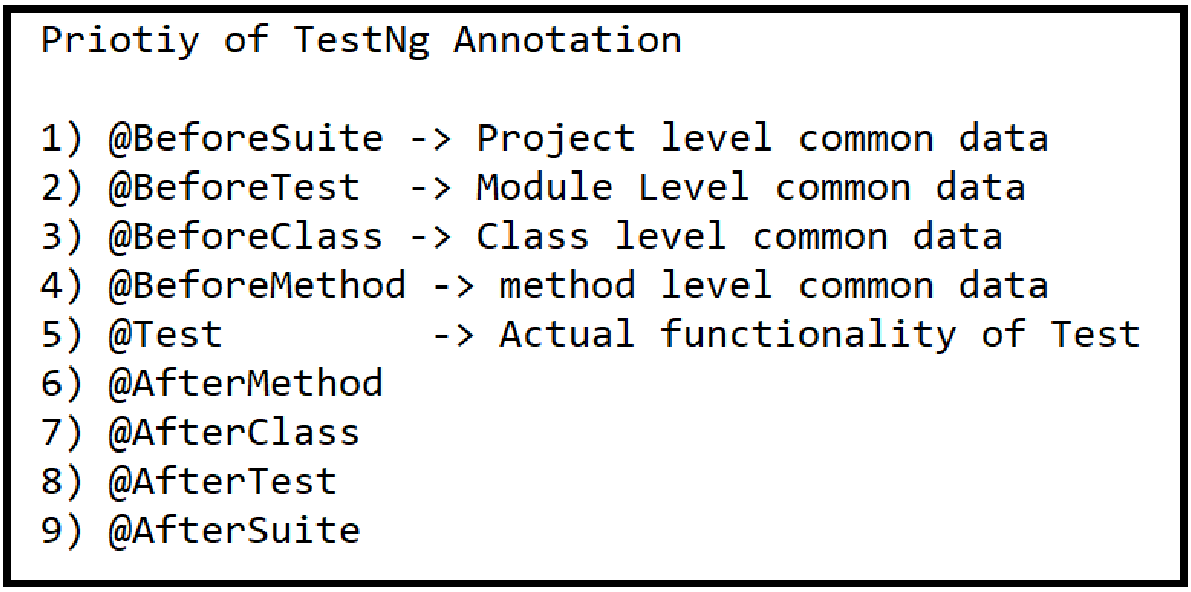
-> If there are multiple methods with @AfterSuite annoation then those methods get executed in Alphabetical order

-> When we should go for @AfterSuite annoation?

Ans: After executing all test cases of all modules then if we want to release all common resource then we will go with @AfterSuite annoation



```
@BeforeGroups
-> It is method level annoation

-> @BeforeGroups annoated methods will be run only one time

-> @BeforeGroups annoated methods run before all the test methods
   belongs to sepeficied group
==================================
@AfterGroups
-> It is method level annoation

-> @AfterGroups annoated methods will be run only one time

-> @AfterGroups annoated methods run after all the test methods
   belongs to sepeficied group
```



```
Priotiy of TestNg Annotation

1) @BeforeSuite -> Project level common data
2) @BeforeTest  -> Module Level common data
3) @BeforeClass -> Class level common data
4) @BeforeMethod -> method level common data
5) @Test         -> Actual functionality of Test
6) @AfterMethod
7) @AfterClass
8) @AfterTest
9) @AfterSuite
```