Automation Testing

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
  cout << "Hello, world!" << endl;
  return 0;
}
```

Disabling Tests

A Day in the Life of an Automation Test Engineer

Marcelo understands and completes his part to write annotation in JUnit and realize the different types of annotations under JUnit 5.

Now, he had to recognize and apply the concept of disabling the annotation on both methods and class for further processing of his project.

This lesson will help him to figure out how to disable the annotation in JUnit.

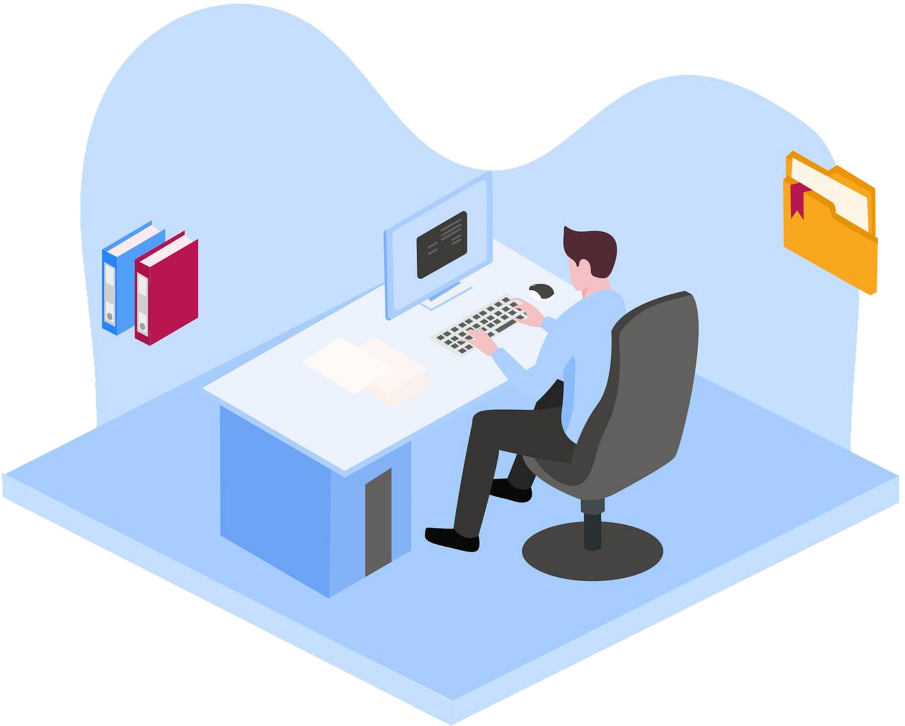
Learning Objectives

By the end of this lesson, you will be able to:

- Understand how to disable annotation on methods
- Analyze how to disable annotation on each method with an example
- Recognize how to disable annotation on class
- Apply concepts to disable annotation on class test with an example

Disabled Annotation on Method

Disabled Annotation on Method



- In JUnit 5, @Disabled annotation can be used to exclude the test methods from the test suite.
- This annotation can be applied over a test class as well as over individual test methods.
- It accepts only one optional parameter, which indicates the reason this test is disabled.

Disabled Annotation on Method

Syntax to disable the annotation on respective method:

```
public class Disabled_MethodTest

  {
    @Disabled("Disabled until Customer Service is up!")
  }
```

Disabled Annotation on Method

Example of disabled annotation on the method:

```
public class Junit5DisableMethodLevelTest {

  @Test
  void test_Add() {
    assertEquals(18, MathUtil.add(7, 11));
    }

  @Test
  void test_Multiply() {
    assertEquals(20, MathUtil.multiple(4, 5));
   }

  @Test
  void test_Devide() {
    assertEquals(5, MathUtil.devide(25, 5));
   }

  @Disabled("Disable until issue #112 got fixed")
  @Test
  void testIs_Prime() {
    assertTrue(MathUtil.isPrime(13));
  }
}
```

Disabled Annotation on Class

Disabled Annotation on Class

- The annotation @Disabled indicates that the annotated test class or test method is now disabled and should not be run.
- When used at the class level, it immediately disables all test methods inside that class.
- A second option would be to disable tests temporarily using the JUnit @Ignore annotation.

Disabled Annotation on Class

Syntax for disable the annotation on a class:

```
@Disabled("Disabled until bug #1423 has been fixed!")

public class Disabled_ClassTest {}
```

Disabled Annotation on Class

Example of disabled annotation on the test class:

```
@Disabled("Disable until issue #1423 got fixed")
public class Junit5DisableClassLevelTest {

  @Test
  void test_Add() {
    assertEquals(5, MathUtil.add(3, 2));
  }
  @Test
  void test_Multiply() {
    assertEquals(15, MathUtil.multiple(3, 5));
  }
  @Test
  void test_Devide() {
    assertEquals(5, MathUtil.devide(25, 5));
  }
  @Test
  void testIs_Prime() {
    assertTrue(MathUtil.isPrime(13));
  }
}
```

Key Takeaways

- In JUnit 5, @Disabled annotation can be used to exclude the test methods from the test suite.
- Disabled annotation can be applied over a test class as well as over individual test methods.
- The annotation @Disabled indicates that the annotated test class is now disabled and should not be run.
- A second option would be to disable tests temporarily using the JUnit @Ignore annotation.